(2)

Natural Language Text Retrieval Using a Large Semantic Network

December 20, 1993

Sponsored by

Advanced Research Projects Agency (DOD)

Defense Small Business Innovation Research Program

ARPA Order No. 5916

Issue by U. S. Army Missile Command Under

Contract # DAAH01-93-C-R201

Contractor: ConQuest Software, Inc.

Business Address: 9700 Patuxent Woods Drive
Suite 140
Columbia, MD 21046

Principal Investigator, Project Scientist, or Engineer: Paul Nelson,
V.P. of Development

Phone Number: 410-290-7150

Effective Date of Contract: 6/29/93
Contract Expiration Date: 1/29/94

Short Title of Work: "Natural Language Text Retrieval"

Reporting Period: Phase 1, Final Report

DISCLAIMER

"The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either express or implied, of the Advanced Research Projects Agency or the U. S. Government."

Natural Language Text Retrieval Using a Large Semantic Network

ARPA Phase I SBIR

Final Report

DAAH01-93-C-R201

Contents

1 Overview

The primary objective of this SBIR effort is to explore and demonstrate advanced performance in precision and recall for text retrieval software. Advanced performance will be accomplished by using dictionaries (specifically WordNet from Princeton University) and other enhancements to the base technology of ConQuest (improved ranking, syntactic processing, etc). Demonstration of accuracy will be performed as a series of retrieval experiments and tests, using the TREC-2 corpus and queries.

The specific objectives for Phase I are as follows:

- **Upgrade the ConQuest dictionary**: The ConQuest dictionary, originally built on WordNet version 1.1 will be upgraded to version 1.4. This necessitates two tasks: 1) Update tools for loading dictionaries so that this upgrade and future upgrades can be accomplished quickly and easily, and 2) Change the dictionary structure to allow for loading multiple dictionaries, which will allow for experimenting with other special purpose dictionaries, such as proper names, abbreviations, acronyms, etc. as time permits.
- **Perform Experiments and Tests**: Queries from TREC-2 will be used to experiment with the new dictionaries and features. Each test will evaluate precision and recall using the standard TREC performance measures. Many tests will be performed, with the intent of demonstrating the gain (or loss) of accuracy for different functions, ranking formulae, and dictionaries.

ConQuest was able to fully accomplish these objectives. Further, the objectives were primarily achieved through software coding (a combination of existing and new software), actual test runs, and detailed data evaluation.

The remainder of this report is divided into the following sections:

- **Dictionaries**: A detailed discussion of dictionary generation, including programs, dictionary sources, dictionary structure, relationships, and evaluation.
- **Text Retrieval Tests**: The results and analysis of running text retrieval tests with and without dictionaries (as well as some other combinations).
- **Other Text Retrieval Functions**: A detailed discussion of a wide range of other text retrieval functions, including an analysis of the like performance improvement (or loss) of each.

2 Dictionaries

2.1 A Brief History of the ConQuest Dictionaries

ConQuest has been working on dictionaries for text retrieval for over four years. The following is a description of six different dictionary revisions which have been generated at ConQuest. Each dictionary revision shows how ConQuest has improved the content, access, and use of the dictionary sources.

1. **The first dictionary** was manually created. It had 500 words, about 1000 links, and only worked for about 3 different queries. It was based on the NL-Builder program, which was used for the very first demonstrations of our natural language based text search ideas.
2. **Several test dictionaries** were generated as intermediate steps towards a truly useful dictionary. These dictionaries were generated as part of the "discovery process" as we learned the structure of the dictionaries, how to access the data, and what the data meant.

   Our knowledge of dictionaries, dictionary loading, the structure of words, etc. was very naive at this point. For example, we did not know that "inflections" or "neighboring terms" existed. Also, we had no idea about the complexities of morphology, or how to combine multiple dictionaries together.

3. **The version 2.5 dictionary** took the experiences we had with the test dictionaries and produced our first broad-based dictionary of English. It had about 40,000 words, and some 80,000 links. It was built using the Webster's Concise Electronic dictionary and thesaurus.

   In terms of words, only the words from Webster's dictionary were loaded. So, words in the thesaurus were not loaded into the dictionaries unless they also happened to be in the dictionary.

   Further, the dictionary did not contain meanings. Or rather, it had meanings from the dictionary, but all the links from the thesaurus were based on the word. There was no way to "choose meanings", because the meanings themselves had no links on them. All links were from word to word.

   Finally, the dictionary only contained synonyms. There were no related terms, antonyms, etc.

4. **A WordNet test dictionary** was used for our semantic network expansion demonstration (where the user double-clicked on a word, and ConQuest identified its meaning based on the word's context).

5. **The initial version 2.7 dictionary** was a huge development effort and saw a synthesis of several technologies and dictionary sources for the first time:
   - Brand-New dictionary Application Programmer Interfaces
   - Full use of Webster's Dictionary and Thesaurus, including all words and link types
   - WordNet words & links
   - Three dictionary sources (Webster's, Thesaurus, and WordNet) merged for the first time
   - Brand-new idiom processing techniques
   - Semantic network topologies
   - Word meanings stored for all words, and links between word meanings
   - Vastly improved morphology algorithms

   The 2.7 dictionaries took over 2 man-years of effort to design and implement.

6. **The version 3.0 dictionary** is essentially the 2.7 dictionary with many cleanups and polishing. This includes the following:
   - Corrected anomalies in the original dictionary (concerning morphology and idiom processing)
   - The dictionary APIs were reworked and polished.
   - Some link topologies were reworked
   - Missing dictionary definitions were automatically generated

These six dictionary revisions show the effort ConQuest has invested in fully understanding the utility of these dictionary resources. This ARPA SBIR contract has been able to leverage off of this understanding to further upgrade the dictionaries and perform many accuracy tests on the information contained within the dictionaries.

2.2 Strengths and Weaknesses of the Dictionary Sources

ConQuest uses three different dictionary resources. These three resources are combined into a single ConQuest dictionary which is used for the full text search and retrieval system. This means that the strengths and weaknesses of the ConQuest dictionaries are primarily derived from the strengths and weaknesses of the original sources.

This section serves to analyze each dictionary resource, and report on the strengths and weaknesses of each.

2.2.1 Webster's Concise Electronic Dictionary

Strengths

- Complete All words have a complete set of meanings, all with definitions, morphological variants, and syntax
- Accurate Information has been review and appears to be very accurate
- Syntactic Features Provides limited features (verb tense, noun plurality)
- Idioms Many useful idioms, especially particles ("roll out", "think up", etc)
- Definitions The most complete and descriptive
- Inflected terms Clearly identifies all morphological variants for all words
- Variant Spellings The only source which contains alternate spellings for words
- Neighboring terms No other source can link "Neighbor" to "Neighborhood"

Weaknesses

- Neighboring terms have no definitions
- Neighboring links and irregular variant links, while useful, do not help semantic processing
- There are very few types of links: Word -> Inflected variation, Word -> Variant Spelling. There are no thesaurus-style links
- Many neighboring terms are idioms. The connection between these words and the original terms is tenuous and unreliable
- Neighboring terms are themselves unreliable. Often the meaning is completely different
- Sometimes words have too many meanings, including very infrequent ones. Example, the word "the" has a meaning as an adverb (as in "the fewer, the better")

2.2.2 Webster's Thesaurus

Strengths

- There are five kinds of links: synonyms, antonyms, see-also, related, and contrasted.
- The first word in each set (i.e. the first synonym in the list, the first antonym in the list) appears to be the most closely related in meaning to the original word. These are the so-called "strong" synonyms
- Where relationships exist, there are lots of them. Often we have 10-20 synonyms for a single meaning of a word
- Contains lots of slang and idiomatic expressions

Weaknesses

- Missing many obvious relationships and words, especially where nouns are concerned. For example, "soup" is related to "mess", but not to "chowder". "computer" is not in the thesaurus at all, but "compute" is.
- Not all associated words are themselves main words in the thesaurus. These words have no direct data for the definition, syntax, or meanings (and so this data must be derived from the source word). For example, "excite" might have "get pumped" as a synonym, but "get pumped" may not be in the thesaurus as a main term.
- Words in the thesaurus are not always root words. Related terms are often past tense verbs, for example.

- There are minimal syntactic features (no noun plurality, and only minimal verb tense information)
- The definitions are brief

2.2.3 Princeton's WordNet

Strengths

- Meanings of words are linked to specific concepts, which are then linked to other meanings of words. This is the only resource which contains such specific and useful information linking meanings of words to other meanings of words.
- The links appear to be of high-quality and relatively well reviewed.
- Most of the words have all of the common interpretations ("soup" has a meaning for "chowder" and "broth", for example).
- The verb frames are potentially useful for future syntax development.
- There are several different types of links between concepts (parent-child, part-of, substance-of, etc).

Weaknesses

- Poor syntax information for nouns and adjectives (only the part of speech).
- Missing adverbs and all close-class words (conjunctions, prepositions, etc.)
- Nouns, adjectives, and verbs are minimally cross-linked. For example there are very few verbs which are linked to nouns.
- Synonym sets in WordNet are a complex topology
- Most definitions are missing or woefully inadequate.
- The dictionary and networks are not as complete or thorough as the proximity sources.
- There are not as many links per word as one would like

The three dictionary sources described above are complementary. Where one is weak, another is strong. Basically, the dictionary is good for syntax information and various word spellings and inflections. The thesaurus is good for verbs and adjectives. WordNet is good for nouns and concept hierarchies.

Therefore, it made sense to combine all three dictionaries into a single resource. This gives ConQuest the most complete and thorough dictionary available.

2.3 Tools for Loading Dictionaries

As part of the ARPA contract, a special tool was designed for loading dictionaries. The "make dictionary" tool normalizes the dictionary loading process and makes it easier to load multiple dictionary formats into a single ConQuest dictionary.

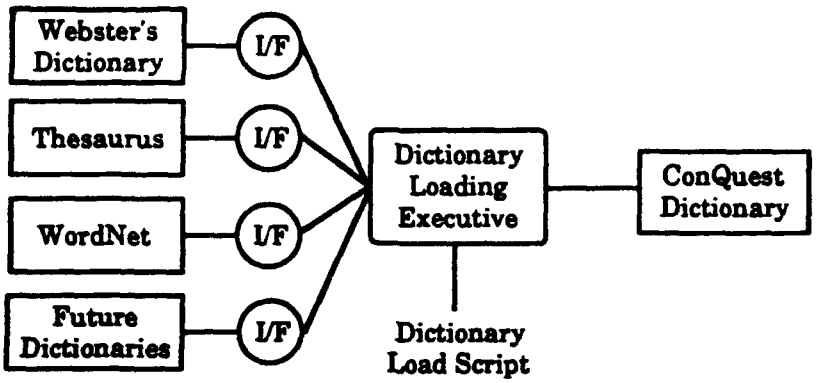


Figure 1 Architecture for Loading Dictionaries

This architecture uses a single interface to communicate between the "Loading Executive" and each dictionary source. This separates the steps of loading dictionaries from the access mechanisms for the multiple dictionary sources. Future dictionaries need only to convert their interface to the standard dictionary loading interface.

This new architecture was used to update the ConQuest dictionary from WordNet 1.1 to WordNet 1.4. Additional evaluations were then performed on WordNet 1.4, the results of which are in section 2.5.

The "Loading Executive" is a set of operations for loading dictionaries. These operations work on all of the dictionary sources in turn (according to the "Load Script") and go through the many steps required to load dictionaries.

The *basic steps for loading a dictionary* are as follows. These steps are followed by the executive when loading dictionaries. The Load Script tells the executive which steps to perform, in what order, and on what dictionaries.

Step 1: Load Main Words and their senses for all dictionary sources

Main words are "root" words such as "neighbor", "run", and "object". These are the un-inflected terms stored in all of the dictionary sources.

Step 2: Load Inflected Words for all dictionary sources

Before adding an inflected word to the dictionary we try and look it up in ConQuest using morphology. If morphology can find the main word, then we skip it (to reduce the dictionary size).

In addition, each inflected word is linked to the main word. This will automatically add inflected terms into the dictionary (such as "caught" for "catch).

Step 3: Load Idioms for all dictionary sources

To improve coverage, all words in an idiom are reduced to root words. For example, the idiom "rolled back" (as in "I rolled back prices") will be stored as "roll back". Because this is the case, idioms must be added after all the atomic words so that morphology can work properly.

Step 4: Load Descriptive Phrases - WordNet

WordNet contains many descriptive phrases as words (called "word collocations in WordNet). These phrases are not really idiomatic expressions, but are a more flexible

combination of the two words. A special placeholder is added to the dictionary for each of these phrases.

For example, "notify_formally" would be added as a single word to the dictionary for the descriptive phrase "notify formally".

Step 5: Load Concepts - WordNet

WordNet has concept nodes, called "synonym sets" or SynSets. Each set contains a list of words which are all synonyms of each other. A placeholder must be added to the dictionary for each SynSet (for all SynSets must be attached to a word).

Step 6: Load all links for all dictionary sources

At this point (now that all words and placeholders have been loaded), links can be added to the dictionary to link these things together. There are four kinds of links:

1. Word to Word links - Neighboring, inflected
2. Meaning to Word Links - Thesaurus links (synonym, antonym, etc.)
3. Meaning to Concept Links - Member of Synset links for Word Net
4. Concept to Concept Links - Between concepts in WordNet

All links are treated pretty much the same by the ConQuest dictionary loading executive.

Step 7: Copy Idiom Info to Irregular Inflected Forms

At this point, ConQuest can find idioms in the dictionary even if the component words are inflected ("run up", "runs up", "running up"). But it can not find idioms where a component word is an irregular inflected form ("ran up", "caught up"). So, idiom information needs to be copied from the regular form to the irregular form.

Step 8: Post Processing Tasks

At this point, all of the raw information has been loaded into ConQuest. What remains is a number of post processing tasks which serve to add additional links for various expansions and to perform dictionary compression.

2.4 Dictionary Structure

ConQuest has split the dictionary into two independent pieces: The words, and the semantic networks. These pieces are stored in separate files (the semantic networks take up two files).

Indexing only requires the word file. Query requires both files, but one at a time (first we look up all the words, then we expand them using the semantic networks. This independence makes for better software maintenance.

Currently, there are two connections between the files:

1. Each meaning of a word in the word file contains a word sense ID value. This ID value points to a node in the semantic net file.

2. The text of the word is stored in every semantic network node. This word is the one which contains the semantic network node as a meaning of the word.

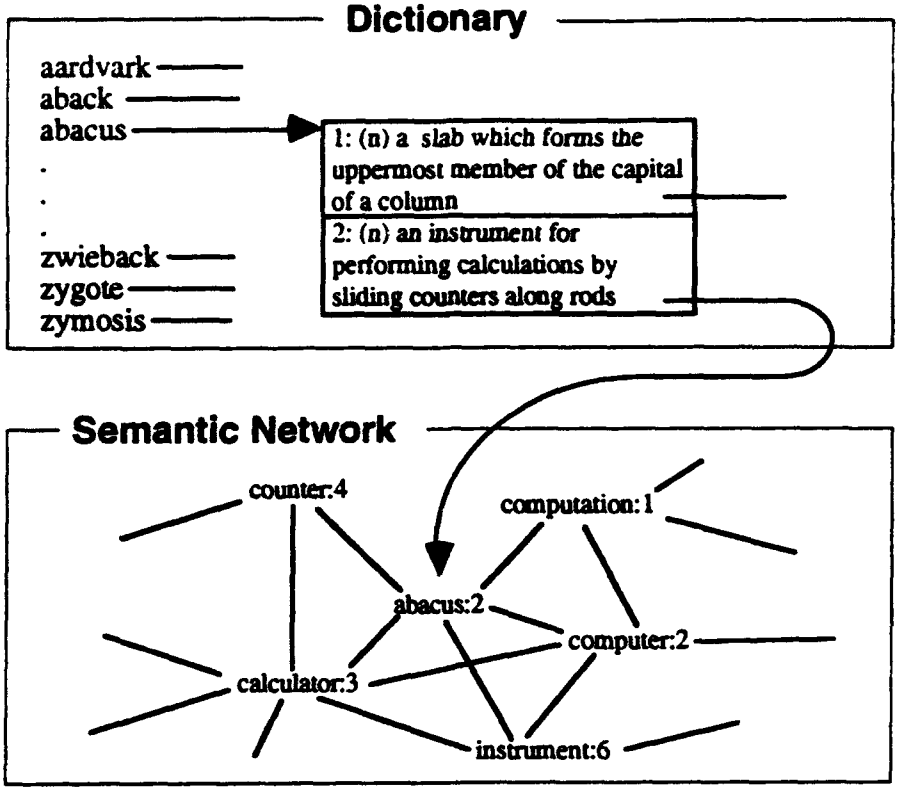


Figure 2 Example Structures for the Dictionary & Semantic Networks

2.4.1 Word Information

Words in ConQuest contain the following information:

1. The text of the word (character string). Note that only root (or reduced) words are stored here, unless
2. A list of meanings in the word

   Each meaning contains the following:

   a. A word sense ID, which is used to index into the semantic networks file
   b. The part of speech (word class) of the word. There are currently 28 different word classes defined in ConQuest. Many are only used for advanced NLP algorithms and are not available from most source dictionaries.
   c. Syntactic feature values. Features include the tense of verbs (present, past, 3rd person singular, past participle), the type of verb (normal, auxiliary), the plurality of nouns (singular or plural), types of adjectives (-est, -er), and other stuff.
3. A "default" meaning for each word. The default meaning is used to represent the word as a whole, as opposed to any particular meaning of the word.

This same information is stored for idioms, which are treated almost exactly the same as words in ConQuest. The only difference is that the information is stored differently, to make the idiom processor more efficient.

2.4.2 Semantic Networks

The semantic network file contains a large set of nodes, and a set of links from each node to other nodes. Nodes in the SNet file are indexed by word sense ID value.

Each node in the semantic network contains the following information:

1. The text of the word which contains this node as a sense of the word
2. A dictionary definition for this node, if available
3. A list of links for the node. Each link contains the following:
   a. The word sense ID of the destination word sense
   b. The dictionary source of the link
   c. The link "relation". This identifies the exact relationship between the two nodes connected by the link. Link relations are different depending on the source:

Dictionary Source	Relations
Webster's Concise Electronic Dictionary	Inflected, Neighboring
Thesaurus	Synonym, Antonym, See-Also, Related, Contrasted
WordNet	Member-of-Synset, Synset-contains-Member, Parent-child, Child-Parent, Part-of, Substance-of, Member-of, Antonym, Similar-to, Wordset-contains-Member, Member-of-Wordset

All links and link types are stored using this basic structure:

Word <-> Word Links (Inflected, Neighboring) - From one default word meaning to another

Meaning <-> Word Links (Thesaurus Links) - From a word meaning to a default word meaning

Meaning <-> Concept Links (Member-of-Synset) - From a word meaning to a concept meaning

Concept <-> Concept (WordNet) - From concept meaning to concept meaning

2.4.3 The Semantic Link Weight Table

The following table is the Semantic Link Weight (SLW) table. This table contains all of the links which are in the ConQuest dictionary, organized as a two dimensional table (dictionary sources across the top, and link relationship types down the left). Each entry shows the name of the link type.

Src:	0	1	2	3
REL	USER	PRX DICT & THES.	WORDNET	CONQUEST
0	32	To Synonym, S>W	Mem Synset, S>C	Word Sense of, S>W
1	-31	To See Also, S>W	Synset CM, C>S	Contains word sense, W>S
2	-30	To Related, S>W	Child of, C>C	To Main Wclass Sense, S>S
3	-29	To Contrasted, S>W	Parent of, C>C	To Sub Wclass Sense, S>S
4	-28	To Antonym, S>W	Antonym, C>C	
5	-27	To Inflected, S>S	Member of, C>C	
6	-26		Substance of, C>C	
7	-25	To Neighboring, W>W	A Part of, C>C	
8	-24		Contains Member, C>C	
9	-23		Contains Subst., C>C	
10	-22		Contains Part, C>C	
11	-21		Mem Wordset, W>S	
12	-20		Wordset CM, S>W	
13	-19			
14	-18			
15	-17			
16	-16	To Strong Synonym, S>W		
17	-15	To Strong See Also, S>W		
18	-14	To Strong Related, S>W		
19	-13	To Strong Contrasted, S>W		
20	-12	To Strong Antonym, S>W		
21	-11			
.				
.				
.				

Figure 3 The Semantic Link Type Map

When ConQuest expands a word using the semantic network, it expands some or all of the meanings. Each meaning of each word is a semantic network node.

The expansion weight always starts out at 100% (this will be the weight of the original meanings of the word). When a link is traversed, the weight is attenuated by the strength of the link. For example, when a link with an 80% strength is traversed, the weight is reduced from 100% to 80%.

However, link strengths are not stored in the semantic networks in ConQuest. Rather, the **link type** is stored, which is a combination of a dictionary source number (0 to 2) and the relationship number (0 to 63). The SLW table above is used to translate from link types to link strengths.

The following shows an SLW table used with ConQuest. This table is loaded when ConQuest is started, and used for term expansions thereafter.

```
/* Default Sal Weight Table */
/* Rel.   Src-0    Src-1   Src-2   Src-3  */
     0       .      80:3    95     128:2
     1       .      80:3   128       0
     2       .      30:3    50:1   110:1
     3       .      10:3    90:1     0
     4       .      50:3    40:1
     5       .     110:1    45:1
     6       .       .      45:1
     7       .      70:3    45:1
     8       .       .      55:1     .
     9       .       .      55:1     .
     10      .       .      55:1     .
     11      .       .       0
     12      .       .      60:3
     15      .       .      55:1
     16      .      95:3    55:1     .
     17      .      95:3   110:1     .
     18      .      40:3   100:1     .
     19      .      20:3    95:1     .
     20      .      60:3     .       .
     32     0:3
     33     4:3
     34     8:3
     35    12:3
     36    16:3
     37    20:3    110:1     .       .
     38    24:3
     39    28:3     80:3     .       .
     40    32:3
     .
     .
     .
```

Figure 4 The Link Strength Table for each Semantic Link Type

The link weights are shown before the colon for each entry. For example, the "Child-Of" link (source=2, relation=2) has a weight of 50 (39%).

The number after the colon is used as a special mechanism for controlling expansions. There are four different values for this number

- 0 Normal link traversal. Expand this type of link whenever the link source is above the expansion threshold. Further, expand the destination of the link, if it is above the threshold.
- 1 Expand once. Do not expand this link if any other link marked as "expand once" has already been traversed.
- 2 Do not expand destination. Add the destination node to the expansion output, but do not further expand the link destination.
- 3 Expand once and Do not expand destination. A combination of #1 and #2 above.

2.5 Dictionary Evaluation

ConQuest has performed extensive evaluations of the dictionary sources which were loaded as the first part of this SBIR contract. These evaluations attempt to quantify the difference between the link relationship types from all of the different dictionary sources. For example, we wish to know the difference in accuracy for the Thesaurus "synonym" link vs the WordNet "SynSet" link.

The steps to dictionary evaluation are: 1) Gain a better understanding of the link topology and expansions using graphs, 2) Create a database of expanded terms, 3) Evaluate the accuracy of each expanded term, 4) Traverse each link individually and accumulate data on the accuracy of its expansions.

Each of these steps is described in the following sections, followed by an analysis of the results.

2.5.1 Expansion Graphs

The dictionary evaluation starts with a better understanding of the links and the expansion process of ConQuest. This is done by performing expansions and plotting graphs which represent the ConQuest expansion process.

A special graph has been created to view the dictionary expansion process. Each link traversed is represented as a row of asterisks. The term weights are plotted across the graph on a scale from 0 to 128, representing 0 to 100%. The asterisks representing the link traversal begin and end at the starting and ending term weights for the link which was traversed.

For example, suppose that a link with a weight of 60 (46%) is traversed. The link will be traversed from some semantic network node to another. If the starting node has a weight of 100 (78%), then the destination node will be achieved with a weight of 46 (35% = 46% * 78%). The graph shows a row of asterisks, representing this link traversal, going from 100 to 46:

```
         120|      110|      100|       90|       80|       70|       60|       50|       40|       30|       20|       10|
    876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543
                           2, 4 *******************************************************
```

Note that the graph starts at "128" (100%), the maximum weight for any term. Also note that the type of link is a "2, 4" link. This stands for dictionary source=2 (WordNet) and link relation=4 (antonym). See section 2.4.3 for a discussion of the link weight table for a description of how link weights using a dictionary source and relation are converted into link strengths.

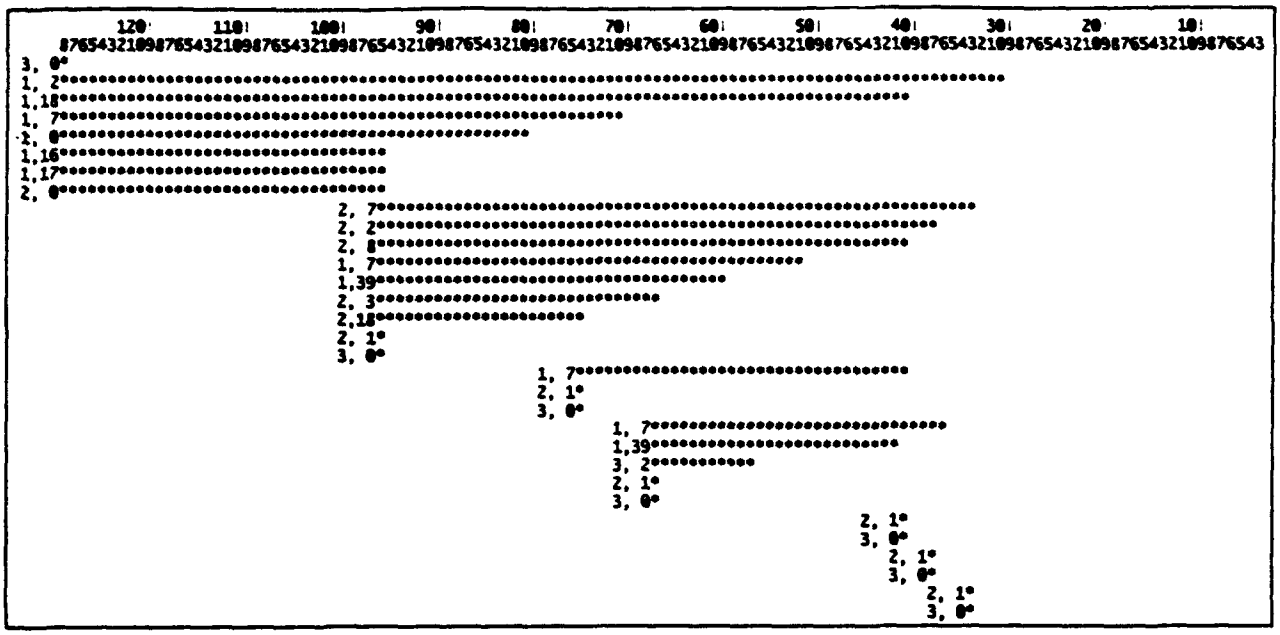
These graphs provide a good view of how expansions occur in ConQuest. For example, here is how a WordNet meaning gets traversed:

```
         120|      110|      100|       90|       80|       70|       60|       50|       40|       30|       20|       10|
    876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543210987654321098765432109876543
2, 0*****************************
3, 0*
                            2, 1*
                            3, 0*
                            2, 2*******************************************************
                                                                                  2, 1*
                                                                                  3, 0*
```

In this example, a wordnet "Member of Synset" (2,0) link is traversed. The strength drops from 128 (100%) down to 100 (78%). Then, a "Child-Of" link is traversed (2,2) which drops the weight down to 46 (35%).

All of the other links have strengths of 128 (100%). This means that they do not reduce the link strength at all. These links include (3,0) = Member of word, and (2,1) = Synset Contains Member.

The following diagram shows the expansions for the word "business" using the standard ConQuest link weight table.



This graph is an extremely useful visual aid to understand better how expansions are performed in ConQuest. Links should be read from left to right. As you come to the end of a link, find out where it picks up by looking straight down, until another link starts (for example, go from link 1,17 to 2,18 to 2,1). These graphs fulfilled three objectives: 1) Gain a better feeling for how ConQuest expansions are performed, 2) Debug the ConQuest semantic networks, and 3) to help design the dictionary evaluation tests.

2.5.2 The Database of Expanded Terms

The second step of dictionary evaluation is to create a databases of expanded terms, and to judge the relevancy of the expanded terms to the original term. The steps for creating this database are as follows:

1. This database is generated by first choosing 100 words for a test database. These 100 terms were taken from a set of 50 test queries. We took words from existing test queries for two reasons: 1) to reduce the chance of a statistical bias in choosing words, and 2) to choose words which users are most likely to query on.
2. Next, meanings were chosen meanings for each of the 100 words. Each of the meanings for each word was inspected, and the meanings most representative of the word in the original query were chosen.
3. Each selected meaning of each word was expanded using the ConQuest semantic network expansion routines. This generated a large database of related terms. For 100 original words, a database of 45,000 related terms were generated (25,000 when all concept nodes and WordNet descriptive phrases were removed).
4. Each related term is now manually judged for its relevancy to the original word. There are five different judgments:
   - 0 **Unrelated** The two terms are not even remotely related
   - 1 **Exact Synonym** The related term is exactly synonymous in meaning to the original query term
   - 2 **Related** The two terms are closely related, but not exact synonyms, "unmarried" to "unattached" would be a good example

3 **Contextual** The related term provides contextual evidence to support the meaning of the query word. For example, "unmarried" to "wedding".

9 **Unknown** Only used for words which have yet to be judged

Using such a system, all 25,000 related terms were manually judged and assigned a relevancy score. The following is an excerpt from the relevancy database:

QUERY: UNMARRIED		QUERY: REGULATOR		QUERY: BOHEMIAN	
SINGLE	1	OFFICIAL	1	NONCONFORMIST	1
DIVORCE	1	BUREAUCRAT	1	BOHEMIAN	1
SEPARATE	1	LEGISLATOR	1	MAVERICK	2
UNWED	1	REGULATE	3	NONCONFORMITY	2
UNMARRIED	1	REGULATORY	3	BEAT	0
SPOUSELESS	1	REGULATIVE	3	RECUSANT	2
SOLE	3	REGULATOR	1	ORIGINAL	2
SINGLY	3	VOLTAGE_REGULATOR	0	ICONOCLAST	2
SINGLENESS	3	PETCOCK	0	HIPPIE	1
SOLENESS	3	STOPCOCK	0	ECCENTRIC	2
SOLELY	3	COCK	0	DROPOUT	2
SOLED	0	WATER_GATE	0	BEATNIK	1
MARRY	3	SLUICEGATE	0	CONFORMER	3
FREE	2	PENSTOCK	0	PEDANT	3
UNFETTERED	1	HEAD_GATE	0	FORMALIST	3
VIRGIN	3	FLOODGATE	0	CONVENTIONALIST	3
UNATTACHED	2	SAFETY_VALVE	0	INDIVIDUALISTIC	2
MAIDEN	2	SAFETY-VALVE	0	UNCONVENTIONAL	2
CELIBATE	2	DRAFT	0	INDIVIDUALIST	2
ATTACHE	0	PEG	0	LAISSES FAIRE	3
WED	3	MONITOR	3	INDEPENDENT-MINDED	2
UNITE	3	GOVERNOR	3	UNUSUAL	2
UNMARRY	2	FLYWHEEL	0	OUT-OF-THE-ORDINARY	2
DISMISS	0	BALLCOCK	0	EXTRAORDINARY	3
PUT	0	BALL_COCK	0	NONSTANDARD	2
DIVORCEMENT	2	APERTURE	0	DEVIATE	2
DISMISSIVE	0	COCKFIGHTING	0	DEVIANT	2
DISMISSAL	0	COCKFIGHT	0	FAR OUT	2
SPLIT	1	DRAFTY	0	WAY-OUT	2
SEPARATE	1	DRAFTINESS	0	OUTLANDISH	2
CANCEL	0	DRAFTILY	0	OFFBEAT	2
ANNUL	1	DRAFTEE	0	NUTTY	2
.		.		.	
.		.		.	
.		.		.	

Figure 5 The Term Relevancy Database

This table provides the necessary data to perform some crude statistical relevancy judgments of term expansions.

2.5.3 Expanding Links and Collecting Relevancy Counts

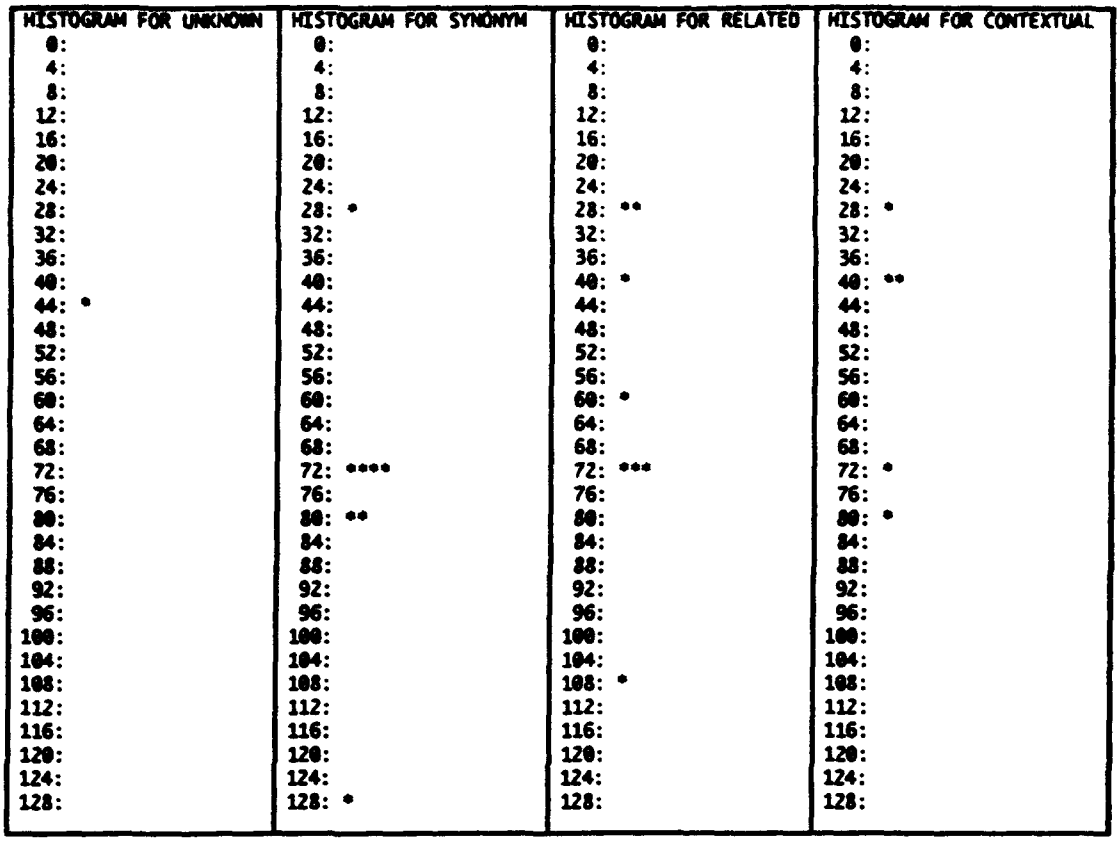
With the database of relevant terms, we can now expand in a variety of ways and test each expansion mechanism. The list of expansions which are created are judged against the database of relevant terms from section 2.5.2.

Three sets of data are generated when evaluating a word: 1) Lists of all related terms which are found, not found, or incorrectly found, 2) Histograms showing the strength of the retrieved terms in each relevancy category (see previous section), and 3) Counts and averages for the word across all retrieved related terms.

A sample output from this process is as follows:

```
Evaluating word UNMARRIED
Statistics for word UNMARRIED:

Found:
 SINGLE/74 UNWED/74 UNMARRIED/128 SPOUSELESS/80 SOLE/80 SINGLY/40 SINGLENESS/40
 FREE/40 UNFETTERED/30 VIRGIN/30 UNATTACHED/74 MAIDEN/30 CELIBATE/30 UNMARRY/110
 UNWEDDED/74 MATELESS/74 UNMATED/81 UNCOMMITTED/74 MARRIED/60 WIDOWED/74
 DIVORCED/74

Not Found:
 DIVORCE/-1 SEPARATE/-1 SOLENESS/-1 SOLELY/-1 MARRY/-1 WED/-1 UNITE/-1
 DIVORCEMENT/-1 SPLIT/-1 SEPARATE/-1 ANNUL/-1 WIDOW/-1 WIDOWHOOD/-1
 UNENCUMBERED/-1 FOOTLOOSE/-1 BREAK UP/-1 FANCY FREE/-1 FREENESS/-1 FREEDOM/-1
 VIRGINITY/-1 VIRGINALLY/-1 VIRGINAL/-1 MAIDENLY/-1 MAIDENHOOD/-1 CELIBACY/-1
 VIRGINALIST/-1 SEPARATENESS/-1 ANNULMENT/-1

Incorrectly Found:
 SINGLE OUT/46

Correctly Not-Found:
 SOLED/-1 ATTACHE/-1 DISMISS/-1 PUT/-1 DISMISSIVE/-1 DISMISSAL/-1 CANCEL/-1
 FREELY/-1 FREEBORN/-1 BREAK/-1 ATTACH/-1 ATTACHMENT/-1 ATTACHABLE/-1 SEIZE/-1
 IMPOUND/-1 CONFISCATE/-1 SEPARATOR/-1 SEPARATIVE/-1 SEPARATION/-1 SEPARATELY/-1
 SEPARABLY/-1 SEPARABLE/-1 SEPARABILITY/-1 CANCELLER/-1 CANCELLATION/-1
 CANCELLABLE/-1 CANCELER/-1 CANCELABLE/-1 BREAK OUT/-1 BREAK OFF/-1 BREAK IN/-1
 BREAK DOWN/-1 BREAKER/-1 BREAKAGE/-1 BREAKABLE/-1 UNITED/-1 ATTACHED/-1

Unknown:

Computing histograms for word UNMARRIED

Number of relevant not retrieved:  28
Number of non relevant correctly not retrieved:  37
```

Figure 6 Related Terms for the Test Word "Unmarried"



HISTOGRAM FOR UNKNOWN	HISTOGRAM FOR SYNONYM	HISTOGRAM FOR RELATED	HISTOGRAM FOR CONTEXTUAL
0:	0:	0:	0:
4:	4:	4:	4:
8:	8:	8:	8:
12:	12:	12:	12:
16:	16:	16:	16:
20:	20:	20:	20:
24:	24:	24:	24:
28:	28: *	28: **	28: *
32:	32:	32:	32:
36:	36:	36:	36:
40:	40:	40: *	40: **
44: *	44:	44:	44:
48:	48:	48:	48:
52:	52:	52:	52:
56:	56:	56:	56:
60:	60:	60: *	60:
64:	64:	64:	64:
68:	68:	68:	68:
72:	72: ****	72: ***	72: *
76:	76:	76:	76:
80:	80: **	80:	80: *
84:	84:	84:	84:
88:	88:	88:	88:
92:	92:	92:	92:
96:	96:	96:	96:
100:	100:	100:	100:
104:	104:	104:	104:
108:	108:	108: *	108:
112:	112:	112:	112:
116:	116:	116:	116:
120:	120:	120:	120:
124:	124:	124:	124:
128:	128: *	128:	128:

Figure 7 Histograms for Each Relevancy Class, by Term Weight

```
SYNONYM Average:  76
NONRELEVANT Average:  46
RELATED Average:  61
CONTEXT Average:  52

SYNONYM count:  8
NONRELEVANT count:  1
RELATED count:  8
CONTEXT count:  5

Total QWord Count:  22
Total Weighed Average:  31
Distance from ideal Average:  32
```

Figure 8 Final Relevancy Totals for "Unmarried"

The above data was for the word "Unmarried" using all of the link in the ConQuest semantic networks. However, it is possible to generate data using only a single link, or just one or two links, by creating new link weight tables. Such a table would have the link weight for all links set to zero, except for the links which we wish to test.

Note that sometimes additional links will need to be set to non-zero values, because they provide transport to the link which is being tested. For example, the "Child-Of" link goes from concept to concept. In order to test this link, the expansion routine first needs to expand from a word meaning to a concept, then through the Child-Of link, and then from the destination concept back to a word meaning.

```
/* Sal Weight Table:  Just Wordnet Child-Of Links */
/* Rel.    Src-0    Src-1   Src-2   Src-3 */
     0        .        .      100     128
     1        .        .      128
     2        .        .        .
     3        .        .        .
     4        .        .        .
     5        .        .        .
     6        .        .        .
     7        .        .       60
     8        .        .        .
     9        .        .        .
     .
     .
     .
```

Figure 9 Weight Table Showing just the "Child-Of" Link

Using table like this, it is now possible to test each link type individually. Since all of the links are turned off except for the one under test, the results of the analysis for each word in the database will indicate the value of the link being tested.

The process for producing the final results is as follows:

1. Load the link weight table to test a single link
2. Collect statistics for all 100 words in the relevancy database
3. Average the statistics across all 100 words
4. Perform the same analysis for the next link type

2.6 Analysis of the Results

The final output from the full dictionary evaluation process is a table of all link types in the ConQuest semantic networks. Each link type has been evaluated individually, and the related terms which can be achieved by the link type have been accumulated.

Four counts are accumulated for each link type:

NR Cnt The average number of Non-Related terms which will be added to the query which this link is traversed. For example, if the number is 1, then traversing the link will add 1 non-related term to the query. If the numbers is ".24", then traversing the link for 100 words will add 24 non-related terms (across all 100).

SY Cnt Average count of synonyms which will be added to the query when the link is traversed.

RL Cnt Average number of related terms

CN Cnt Average number of contextual terms

The results from the dictionary evaluation are shown in the following table:

Link Type	Source	NR Cnt	SY Cnt	RL Cnt	CN Cnt	Total Cnt	% Rel	Wgt % Rel	Rank
see also	Thesaurus	0	0.01	0.01	0	0.02	100%	83%	1
strong see also	Thesaurus	0	0.06	0.06	0.01	0.13	100%	79%	2
rev inflected	Dictionary	0	0.01	0.03	0	0.04	100%	75%	3
see also	WordNet	0	0.03	0.03	0.02	0.08	100%	71%	4
synset	WordNet	0.24	0.74	0.54	0.26	1.78	87%	67%	5
similar to	WordNet	0.02	0.08	0.15	0.06	0.31	94%	65%	6
strong synonym	Thesaurus	0.11	0.28	0.26	0.13	0.78	86%	64%	7
rev neighboring	Dictionary	0.01	0.01	0.09	0.01	0.12	92%	61%	8
contains part	WordNet	0.02	0.02	0.19	0.03	0.26	92%	60%	9
synonym	Thesaurus	0.69	1.08	1.11	0.39	3.27	79%	60%	10
contains mem	WordNet	0	0.03	0.03	0.08	0.14	100%	55%	11
neighboring	Dictionary	0.31	0.05	0.82	0.26	1.44	78%	47%	12
part of	WordNet	0.01	0	0.04	0.03	0.08	88%	46%	13
parent of	WordNet	2.43	0.91	2.83	1.23	7.4	67%	43%	14
antonym	WordNet	0.04	0	0.1	0.06	0.2	80%	43%	15
related	Thesaurus	1.7	0.58	1.29	0.63	4.2	60%	39%	16
strong antonym	Thesaurus	0.06	0.01	0.08	0.08	0.23	74%	39%	17
child of	WordNet	0.83	0.36	0.35	0.31	1.85	55%	38%	18
strong related	Thesaurus	0.24	0.05	0.21	0.11	0.61	61%	37%	19
inflected	Dictionary	0	0	0	0.01	0.01	100%	33%	20
pertains to	WordNet	0.04	0	0.04	0.02	0.1	60%	33%	21
antonym	Thesaurus	0.01	0	0.01	0.01	0.03	67%	33%	21
strong contrasted	Thesaurus	0.18	0	0.04	0.07	0.29	38%	17%	23
contrasted	Thesaurus	0.92	0	0.09	0.31	1.32	30%	12%	24
causes	WordNet	0.02	0	0	0	0.02	0%	0%	25
attribute	WordNet	NOT IN TEST SUITE							25
contains sub	WordNet	NOT IN TEST SUITE							25
derived from	WordNet	NOT IN TEST SUITE							25
member of	WordNet	NOT IN TEST SUITE							25
requires	WordNet	NOT IN TEST SUITE							25
substance of	WordNet	NOT IN TEST SUITE							25
wordset contains	WordNet	NOT IN TEST SUITE							25
		NR Cnt	SY Cnt	RL Cnt	CN Cnt				
Weighting:		0	3	2	1				

Figure 10 Dictionary Evaluation Results, Rankings, and Suggested Table Weights

There are four columns above which require additional explanation:

Total Cnt	A sum of "NR Cnt", "SY Cnt", "RL Cnt", and "CN Cnt"
% Rel	(NR_Cnt + SY_Cnt + RL_Cnt + CN_Cnt) / Total_Cnt
Wgt % Rel	(3*SY_Cnt + 2*RL_Cnt + CN_Cnt) / (3*Total_Cnt)
Rank	A ranking of each link type based on "Wgt % Rel"

The most interesting of these numbers is "Wgt % Rel" which is an attempt to give a weighted relevancy score to each link. The percentage is the probability that the user will get a synonymous meaning (where related terms count for 2/3 and contextual terms count for 1/3). The "Wgt % Rel" was used to specify the strengths of each link in the Semantic Link Weight Table. These numbers will then be passed to the ConQuest search engine and used for statistical text retrieval and ranking.

The first four link types in the table (all of them with a "% Rel" of 100%) should be discounted. The "Total Cnt" for these links is very low, meaning that very few word meanings had this type of link to be traversed. The same is true of the last seven link types in the table, none of which were ever traversed by the test suite which was generated. These are rare links in the WordNet database.

Performing this style of analysis (i.e. ignoring link types which minimal data in the test database), we see that WordNet synonyms and Thesaurus synonyms are near the top of the list, as expected. Further, links which you would expect to be weaker (such as contrasted terms, related terms, etc) are truly ranked lower.

Finally, there does not seem to be any source which is noticeably better or worse than the other. All of the different link types from all sources appear to be intermixed.

2.7 "Multi-Layered" Dictionaries

The final task regarding dictionary development was to explore methods for loading specialized dictionaries into ConQuest. Such dictionaries include vertical domains (Legal, Medical, etc), corporate dictionaries (to hold product names and industry specific jargon), group dictionaries (for a particular project), and personal dictionaries (for the user's own dictionary terms). These could also include abbreviations, proper names, geographical locations, and so on.

These dictionaries would be arranged into a hierarchy of dictionaries:

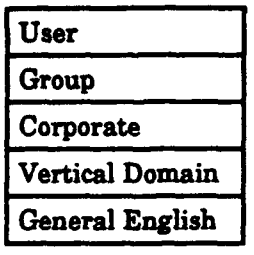


Figure 11 Weight Table Showing just the "Child-Of" Link

This architecture is called "Multi-Layered" dictionaries. Each layer serves to handle the terms for a particular user or user group. All layers are stored in separate files, so that each dictionary can be edited without affecting the others.

When ConQuest goes to look up a term in a multi-layered dictionary, it starts with the highest dictionary, the user's dictionary. If the term is not there, then it goes down and checks the group's dictionary. This continues until, finally, the General English dictionary is checked. The meanings from all the dictionaries where the term exists will be combined together to represent the complete knowledge of the word.

After combining the meanings from all dictionaries, the word is presented to the user. Depending on the interface, all meanings may be presented to the user, or only selected meanings (for example, only meanings from the user's personal dictionary, if they exist).

ConQuest was able to complete a prototype version of the multi-layered dictionary concept. The following functions were demonstrated:

- Installing multiple dictionaries
- Looking up words and returning all meanings from all dictionaries
- Performing expansions which cross dictionary boundaries

These simple demonstrations have been enough to prove that the concept is feasible, and that the access time for a word in this architecture is reasonably fast (up to 30,000 words per minute on an IBM 486 PC Clone running at 33 Mhz) for full text indexing and retrieval.

However, the prototype we generated was only for demonstration. Additional work will be required before a full set of multi-layered dictionaries can be installed and tested.

3 Text Retrieval Tests

As part of this SBIR Phase I contract, ConQuest has performed a series of tests to show the value of adding the dictionary with semantic networks for text retrieval accuracy. The purpose of section 3 is to describe these tests and to report on the results.

Due to the limited time and disk space available for this SBIR Phase I, ConQuest has performed these text retrieval tests on a sub-set of the TREC-2 database, consisting entirely of Wall Street Journal texts. The texts were taken from disks 1 & 2 of the TREC database and totaled 402 Mb of data. The indexes built for queries totaled 211 Mb (about 52% of the sizes of the raw text database).

3.1 Query Generation Process and Tools

The first step in performing a text retrieval test is to generate the queries which are going to be used for testing. The ConQuest queries were generated with a combination of automatic and manual processes for TREC-2. The same queries which were submitted to TREC-2 were used for most of the tests were used for the test results shown below.

Testing text retrieval systems is a very touchy process, so it is important to generate good queries. A good query will reduce the amount of noise a system will have to deal with, and generally allow for a finer and more detailed analysis of the results.

This is why ConQuest has opted to manually edit the TREC-2 queries. The purpose is to remove words, meanings, and expansions which will not aid the query, so that the remainder will provide more accurate results of different test strategies. We are striving to remove irrelevant terms which would only serve to clutter and muddy the results.

However, since query generation is partly manual, it is important to understand the process to verify that the human intervention has not biased the search results.

The steps for generating a query are as follows:

1. Automatically convert the TREC topic descriptions into query log files.
2. Manually remove irrelevant words, meanings of words, and word expansions
3. Group words into conceptual sets (done for a special retrieval test)

Each of these steps is described in the following sub-sections.

Note that all steps were performed above for all queries before any text retrieval tests were executed. No queries were performed against a database, and no feedback information was used in generating the queries. This makes ConQuest fully compliant with the rules for ad-hoc queries in TREC-2.

3.1.1 Converting TREC-2 Topic Descriptions to Query Log Files

A special program has been created to convert TREC-2 topic descriptions into ConQuest query log files. The architecture of this program is show in figure 12.

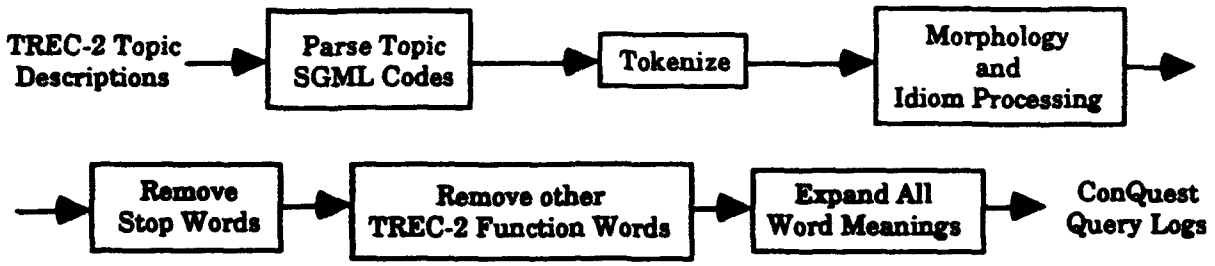


Figure 12 Program to Automatically Generate Query Log Files

The modules in the program are as follows:

Parse Topic - Reads through the topic looking for the SGML codes (such as <description>). The location within the topic for all words in the query are preserved in the final query log files.

Tokenize - Divides up strings into tokens. The tokenizer was modified somewhat for TREC-2 to handle tokens like AT&T (specified as AT&T in SGML notation).

Morphology - Locates all words in the dictionary and reduces them to root words if possible.

Idiom Processing - Collects idioms together as single terms, such as "United States".

Remove Stop Words - Removes conjunctions, determiners, auxiliary verbs, prepositions, etc.

Remove Function Words - Removes words such as "document", "relevant", and "retrieve" which are often used in TREC-2 narratives but do not aid retrieval at all.

Expand Word Meanings - All word meanings are expanded using the ConQuest semantic network and all expansions are added to the query.

This program also generates other statistics, such as the count of each term in the query, a count for each term for each section of the query (sections being the topic, description, narrative, concepts, and factors), and the total number of words in the query.

```
WORD_INFO :
  WORD = "PHILIPPINES"
  TREC = 99 ( 0 0 0 0 1 0 UNDER "COUNTRIES" )

WSENSES_INFO :
  ID = 203989
  DEF = "B C-PHILIPPINE_ISLANDS, REPUBLIC_OF_THE_PHILIPPINES"
  EXPANSION :
    DEST_WSID = 53559 DEST_TEXT =  "PHILIPPINES" EXPANSION_WEIGHT = 128
    DEST_WSID = 203985 DEST_TEXT =  "_PHILIPPINE_ISLANDS" EXPANSION_WEIGHT = 95
    DEST_WSID = 53563 DEST_TEXT =  "PHILIPPINE" EXPANSION_WEIGHT = 70
    DEST_WSID = 53561 DEST_TEXT =  "FILIPINO" EXPANSION_WEIGHT = 70

  ID = 203990
  DEF = "A C-REPUBLIC_OF_THE_PHILIPPINES"
  EXPANSION :
    DEST_WSID = 53559 DEST_TEXT =  "PHILIPPINES" EXPANSION_WEIGHT = 128
    DEST_WSID = 210929 DEST_TEXT =  "_REPUBLIC_OF_THE_PHILIPPINES" EXPANSION_WEIGHT = 95
    DEST_WSID = 53563 DEST_TEXT =  "PHILIPPINE" EXPANSION_WEIGHT = 70
    DEST_WSID = 53561 DEST_TEXT =  "FILIPINO" EXPANSION_WEIGHT = 70
```

Figure 13 An Example of a Query Log File for the Word "Philippines"

3.1.2 Manually Remove Words, Meanings, and Expansions

The next step in query generation is to manually remove words, meanings, and expansions from the Query Log file. Fortunately, ConQuest has graphical user interfaces (GUIs) for performing this task with our commercial product, and so the task proceeds quickly. When these terms are removed, it is indicated in the log file with a semi-colon before the item which was removed. This allows for later review of this manual process.

Removing irrelevant terms is important to improve the accuracy of the statistical analysis. Irrelevant terms can only serve to add noise to a search. That is, irrelevant documents containing these terms would be retrieved and clutter up the search set. The statistics for such a set would then be lower (i.e. fewer relevant documents retrieved). Such statistics would have a greater percentage of error.

However, to eliminate any human bias to the test results, no document feedback is used for this manual step. All query adjustments are performed without actually executing the query. Only until all queries have been generated and the query log files are fixed will the queries be used for tests and evaluations.

3.1.3 Group Words into Sets

The final step in query generation is to group words in each query into sets which represent the main concepts of the query. This grouping can be used with special statistical processing to improve the accuracy of the queries.

Some word grouping is done automatically, namely the grouping of expansions underneath the word from which they were derived. For example, if "charge" is expanded to "blame", "indict", and "accuse", these three expansions are grouped with the word "charge". This grouping is used to perform a level of contextual evidence processing for the words.

To perform contextual evidence ranking, ConQuest applies a different statistical function among words in a query as opposed to within the set of expansions for a word. Basically, the function within the expansions is more "OR-like" (the accrue function) as opposed to "AND-like" (the average function). There is still much debate about what these functions should actually look like. For now, the Accrue() and Average() functions seem to work the best.

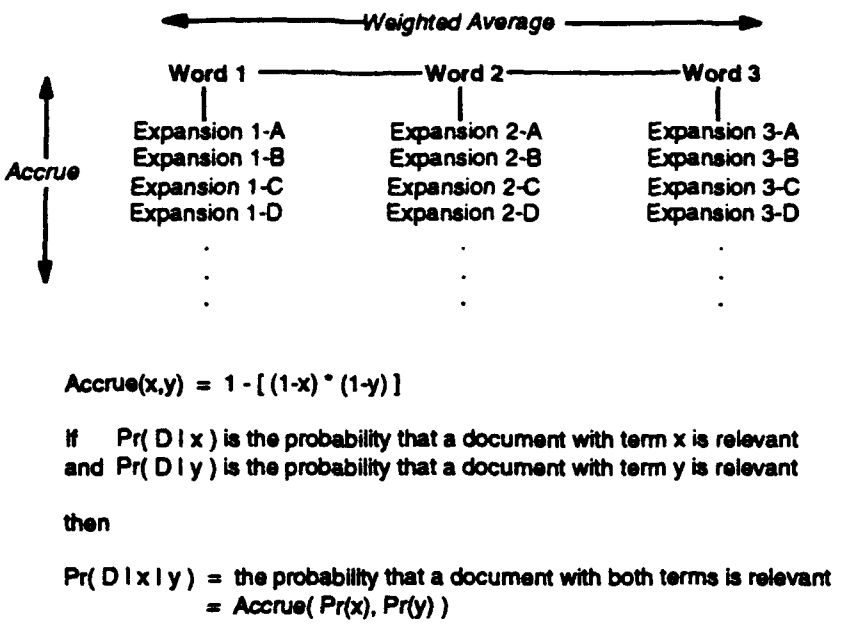


Figure 14 Ranking Functions Among Words and Within Word Expansions

When words are manually grouped into sets, the Accrue() function will be applied to all the words in the set. This is useful for many of the TREC-2 queries, because they often give lists of terms which are examples of a particular concept. All of these examples can be grouped together.

Another advantage to grouping is that it helps improve the accuracy by forcing the search system to choose at least one term from each main concept. The grouping identifies the main concepts, and the Average() function will automatically prefer to take a term from each group.

Finally, the performance improvement from manually grouping terms was the subject of a special accuracy test. The results on this test are shown in section 3.3.

3.1.4 Review the Queries

A special tool has been created to review the queries. This tool provides a quick display of the terms in the query so that they can be quickly reviewed for relevancy to the query.

```
MAIN WORD:  LABORATORIES x1 (0x:1#:128%:99)
    LABORATORY (128%) LAB (95%) LABORATORY (128%)

MAIN WORD:  BIOTECHNOLOGY x1 (1x:19#:128%:99)

MAIN WORD:  FINANCE x1 (0x:1#:128%:99)
    CONTAINS:  VENTURE CAPITAL x1 (0x:2#:128%:0)
    CONTAINS:  CAPITAL x1 (0x:1#:128%:99)
    CONTAINS:  VENTURE x1 (0x:1#:128%:99)
    CONTAINS:  STOCK x1 (0x:1#:128%:99) STOCK (128%) _PREFERRED_STOCK (66%)
               _PREFERENCE_SHARES (66%) _ORDINARY_SHARES (66%) _COMMON_STOCK (66%)
               BLUE CHIP (66%) _BLUE-CHIP_STOCK (66%) _CAPITAL_STOCK (66%)
```

```
CONTAINS:  CAPITALIZATION x1 (0x:1#:128%:99) CAPITALIZATION (128%) CAPITALIZE
           (80%) CAPITALIZATION (128%) CAPITALIZE (80%) CAPITALIZATION (128%)
           CAPITALIZE (80%) CAPITALIZATION (128%) CAPITALIZE (80%) CAPITALIZE
           (80%)
CONTAINS:  INVESTMENT x1 (3x:3#:128%:99) INVESTMENT (128%) VENTURE (66%) INVEST
           (80%)
CONTAINS:  INVESTED x1 (3x:1#:128%:99) INVEST (128%) COMMIT (95%) INVESTOR (70%)
           INVESTMENT (70%) INVESTABLE (70%) SPECULATE (66%) INVEST (128%)
           INVESTOR (70%) INVESTMENT (70%) INVESTABLE (70%) INVESTOR (70%)
           INVESTMENT (70%) INVESTABLE (70%)
CONTAINS:  FINANCIAL x1 (3x:2#:128%:99)
CONTAINS:  FUNDING x1 (2x:2#:128%:99)
```

Figure 15 Output from Query Review Program

The output above shows several statistics for each word. For example, "x1 (3x:3#:128%:99)". The following describes the purpose of this cryptic output:

- **x1** Gives the multiplier assigned to each word. Typically, a multiplier is computed automatically based on a statistical evaluation of the words. For the purposes of this contract, all multipliers were set to one (1).
- **(3x** The multiplier assigned by a human to indicate the strength of the word. This is ignored for the purposes of these SBIR tests.
- **3#** The number of occurrences for the word in the TREC Topic description.
- **128%** The weight of the expanded term from the semantic network (128 stands for 100%)
- **:99** Originally intended to hold the concept number if the term occurred in the <concepts> section of the TREC-2 topic description, but unused now.

3.2 Text Execution Process and Tools

Once the query log files have been created, a series of tests and evaluations can be executed. The architecture for this phase is shown in figure 16.

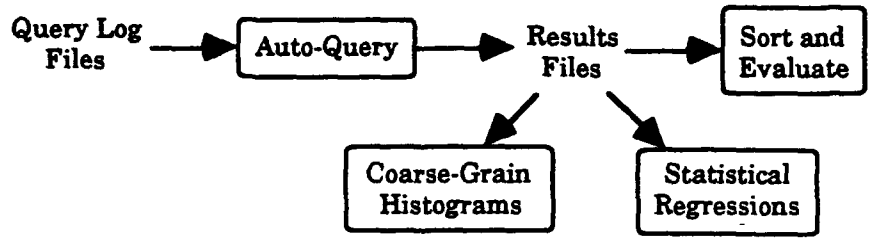


Figure 16 Auto Query and Results Generation

Each of these programs is described in the following sections.

3.2.1 Auto Query

The "Auto Query" program is responsible for executing all queries from query log files (see figure 15) and creating the results files. The results files have sufficient statistical information to perform a variety of analyses.

```
106 Q0 WSJ911101-0173    0  334    0  742   23  218 CnQst -
106 Q0 WSJ861203-0047    1  256    0  742   18  192 CnQst +
106 Q0 WSJ870422-0076    2  256    0  742   52  239 CnQst +
106 Q0 WSJ890731-0115    3  491  454  736   16  332 CnQst -
106 Q0 WSJ910920-0131    4  472    0  736   21  308 CnQst -
106 Q0 WSJ870624-0082    5  256    0  736   12  237 CnQst +
106 Q0 WSJ870610-0014    6  256    0  728    6  245 CnQst ?
106 Q0 WSJ870717-0152    7  491  706  728   15  314 CnQst +
106 Q0 WSJ871026-0024    8  256    0  727   11  144 CnQst -
106 Q0 WSJ880628-0149    9  472    0  727   24  315 CnQst -
```

Figure 17 Results File from Auto Query

The result file originally started out being the same as the result files for TREC-2, but then has evolved to contain some additional statistics.

The columns of this result file are as follows:

1.	"106"	The TREC-2 query ID
2.	"Q0"	Always "Q0", ignored by all programs
3.	"WSJ911101-0173"	The TREC-2 document ID
4.	"0"	The rank of the document (a counter from 0 to 4999)
5.	"334"	Maximum single hit within the document
6.	"0"	The thresholded document hit rank
7.	"742"	The coarse-grain rank
8.	"23"	The number of hits within the document
9.	"218"	The average strength of the hits in the document
10.	"CnQst"	Always "CnQst", ignored by all programs
11.	"-"	The relevancy of the document ("+" = relevant, "-" = known non-relevant, and "?" = not judged)

The Auto Query program has several different options for executing queries, each intended to test a different aspect of the search software. All of the functions shown below are performed automatically. The original query log files remain untouched.

Remove Meanings - Remove meanings from selected dictionaries from all words in the query. This feature is used to test how the system behaves with only the WordNet or Thesaurus dictionary sources.

Remove Words - All words in the query log file identify from where in the TREC topic description they originally came from (for example, from the topic, description, narrative, concepts, or factors sections). Words can be removed based on which sections of the topic description are desired.

Remove Expansions - A primary purpose of the dictionaries in ConQuest is to provide a way to semantically expand the words in a query. If all the expansions are removed, it is as if there were no dictionary in the system at all. So, removing expansions is intended to test how the system behaves without a dictionary, to see how much the dictionary improves the search performance.

Term Weighting - Several functions are available for various kinds of term weighting based on term statistics (such as query word frequency, database frequency, etc.). However, this was not a major focus of this SBIR Phase I contract, and so all terms were weighted equally for the tests shown below.

Collecting Word Sets - Word sets are collected together based on the data identified by the query editor in the query log file. Using this function, the accuracy of creating these word sets (as opposed to every word being independent) can be evaluated.

3.2.2 Query Sorting and Formatting

Once the results files have been generated by Auto Query, they can be sorted and reformatted for evaluation by the TREC-2 standard evaluation metrics.

Figure 18 Producing the Final TREC-2 Evaluation Data

The ConQuest processing occurs in two steps. The first step is the "coarse-grain" rank. This is the step actually performed by the retrieval engine. This ranking is integrated with retrieval so that a special searching process can be employed to find the best documents first (that is, the best based on the coarse-grain ranking function). Essentially, coarse-grain rank considers the absence or presence of each query word in the document. It does not consider the frequency of the words, or the proximity or location of the words in the document itself.

After doing coarse-grain rank, the hits (highlights) for a document are retrieved and additional statistics are created. This includes the Maximum Hit for the document, the Average Hit Strength, and the Thresholded Document Hit Rank. All of these statistics include document word frequency and query word proximity as factors.

All of these statistics are provided to the query computation and sorting functions shown above. The "Final Fine-Grain" computation takes these statistics and uses them in a formulae to produce the best overall document score. The Sort function sorts by this Final Fine-Grain score. Then, the top 1000 documents are passed to the TREC-2 Evaluation program.

3.2.3 Histogram Generator

One program used to interpret results is a histogram generator. This program tabulates the results from Auto-Query and produces a two-dimensional table of relevant document counts. Relevant documents are counted across each 250 documents (up to 5000 total documents retrieved), and for each query in the database.

Figure 19 shows a histogram for query number 110. The bars represent the number of relevant documents found in each 250 document range. For example, the first bar shows that 70 relevant documents were found in the first 250 documents retrieved.

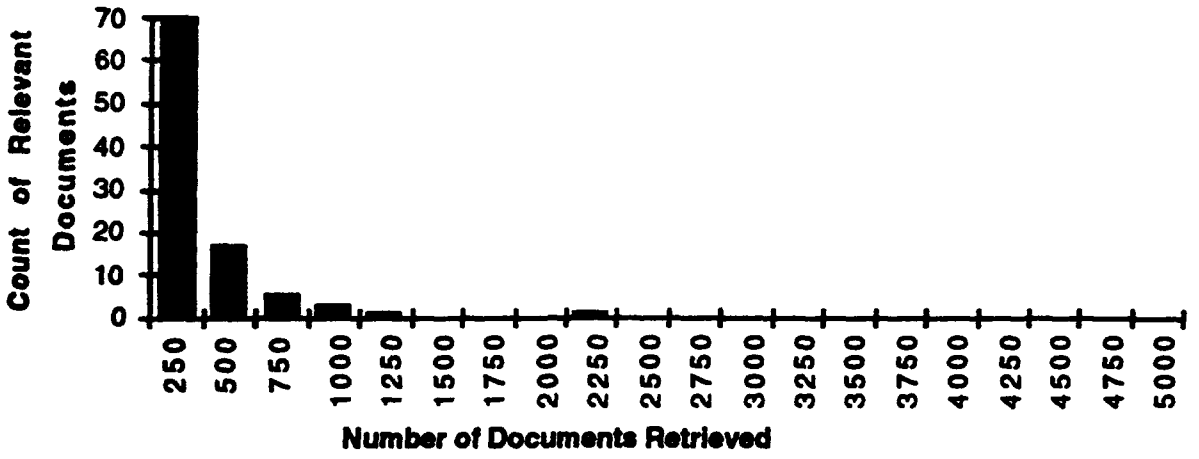


Figure 19 Histogram of Relevant Documents Retrieved for Query #110

Note that this histogram shows the results for coarse-grain ranking only. Typically, the results will show much better performance when using additional statistical analysis coupled with the final fine-grain sorting analysis.

The following figure (number 20) shows a histogram for all queries in the TREC database for the first 250 documents (based on coarse-grain rank). Essentially, each bar shows a count of the number of relevant documents for each query ID in the TREC-2 query set, after the first 250 documents have been retrieved.

Using these histograms, ConQuest can compute the number of documents which must be retrieved by ConQuest in order to get a possible recall percentage. If the possible recall percentage is too low, then more documents (more than 5000) could be retrieved. If the percentage is high enough, then fewer documents could be retrieved (say, 2000) without a loss of overall accuracy.

The results of this analysis help identify the performance of the coarse-grain ranking and retrieval algorithms.

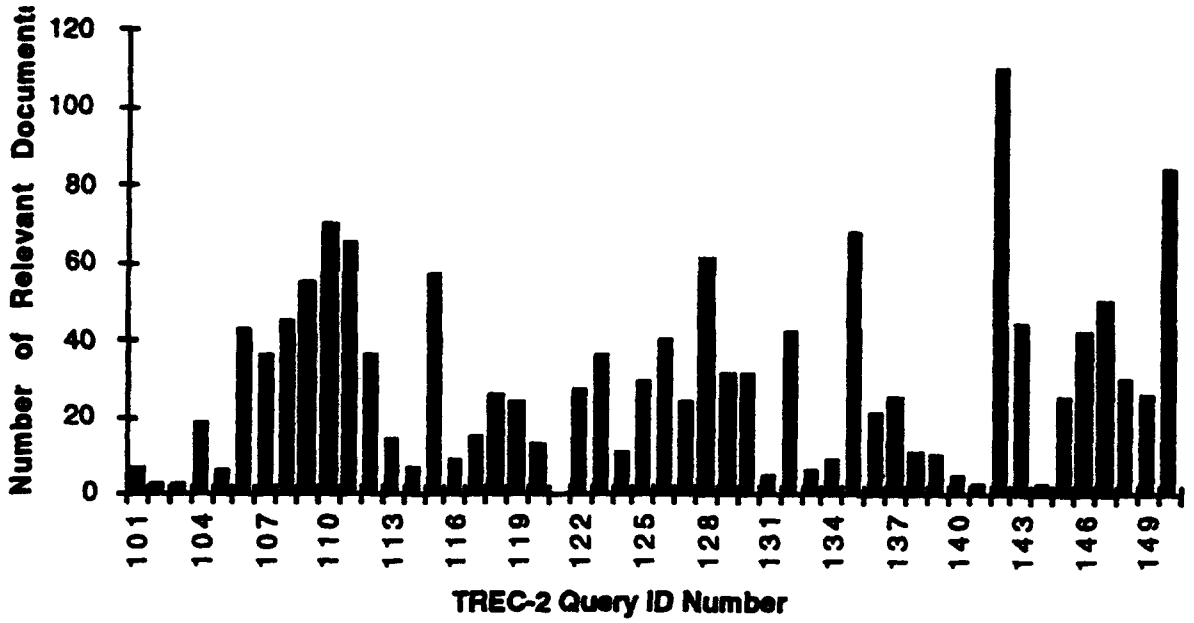


Figure 20 Histogram of Relevant Documents Retrieved for All Queries

3.2.4 Query Relevancy Statistical Analysis

ConQuest has been experimenting with statistical regressions to determine the best combination of ranking statistics stored in the results files (see section 3.2.1). While this analysis is not the primary focus of this SBIR Phase I, this section (3.2.4) covers some preliminary results which may help guide a Phase II effort.

The statistics from the results file are used as predictors for a statistical regression. The desired output is the relevancy judgment (0 or 1). Using these techniques, a slightly better retrieval graph can be achieved.

For example, the following graphs show the result of performing statistical regressions on the results from query #135. The number of documents retrieved are shown along the X-Axis. The cumulative total of relevant documents are on the Y-Axis. For example, at 500 documents retrieved, the cumulative total is about 72 documents for coarse grain sorting and 79 documents for fine-grain sorting. This means that 72 of the previous 500 documents were judged to be relevant. The maximum possible number of relevant documents is 83 documents.

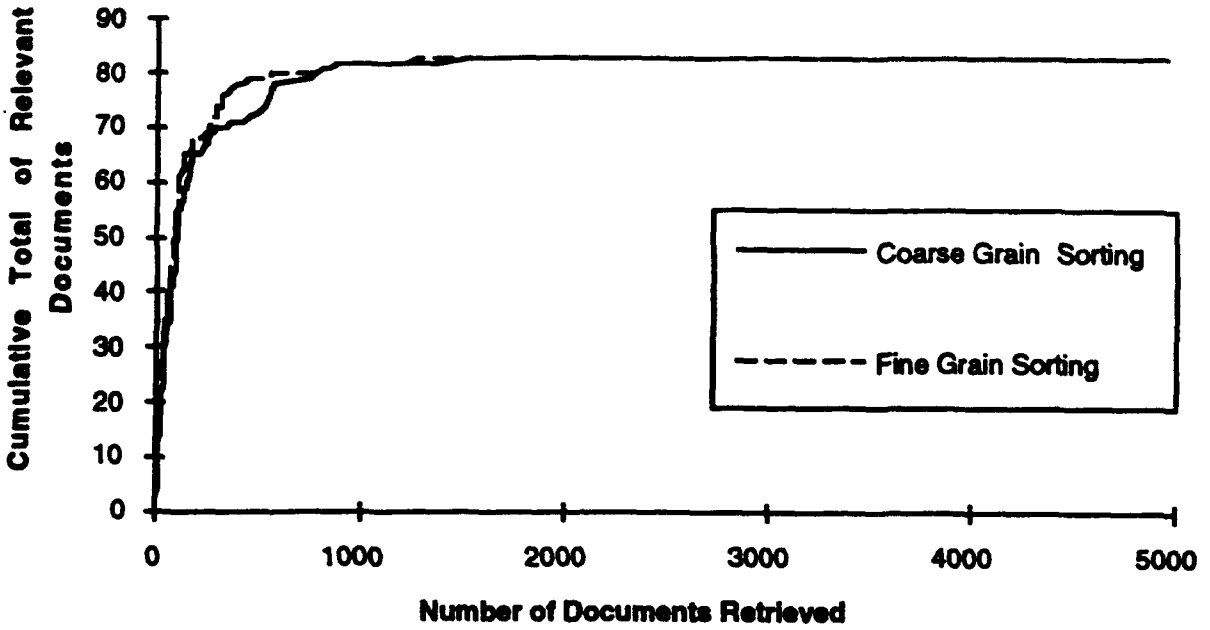


Figure 21 Coarse Grain Sorting vs Fine Grain Sorting

This graph shows only a slight difference between the coarse grain and fine-grain sorting, about 10% at most. We believe there are two primary reasons for this small difference:

1. The statistics for fine-grain ranking need to be improved. This includes the maximum hit and average hit statistics. There are several improvements to the algorithms which should behave better for TREC-2 queries which typically have many dozens of words (the original statistics were tuned for very small queries).
2. The statistics for each word of the query need to be separated out for better term weighting. Currently, all statistics operate over all of the words in the query, and attempt to average all word statistics into 4 or 5 numbers per document. If the statistics for each individual word are separated out, then regressions more like those reported in the TREC-1 conference proceedings can be performed.

This SBIR Phase I contract is primarily concerned with coarse-grain sorting algorithms and the affect of dictionaries. However, these statistical analyses highlight the need for more work on fine-grain sorting.

3.2.5 Cumulative Recall Percentage

The tests in this section all use coarse-grain ranking as the sole ranking algorithm. As noted in the previous section (3.2.4), fine-grain ranking is based in part on statistical regressions which need a more thorough and detailed analysis.

All of the graphs in this section show results in terms of Cumulative "Recall Percentage". This number shows the total percentage of relevant documents retrieved at any point in a list of documents retrieved.

For example, if 50 relevant documents were retrieved in the first 250 documents, and there are 200 total relevant documents, the recall percentage at 250 would be 25%. If another 50 relevant documents are retrieved in the next 250 documents, the recall percentage at 500 would be 50%.

Total Documents Retrieved	250	500	750	1000	1250	1500	1750	2000	2250	2500
Number Relevant in Range	63	11	8	1	8	1	0	1	0	0
Cumulative Relevant	63	74	82	83	91	92	92	93	93	93
Cumulative Recall Percent	65%	76%	85%	86%	94%	95%	95%	96%	96%	96%

2750	3000	3250	3500	3750	4000	4250	4500	4750	5000
0	4	0	0	0	0	0	0	0	0
93	97	97	97	97	97	97	97	97	97
96%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Figure 22 Relevancy Statistics for Query #101 (Full system, with dictionary)

In the example above, there is a maximum of 97 possible documents to retrieve. This is why the cumulative recall percentage grows steadily, and reaches 100% after 97 relevant documents have been retrieved.

When producing graphs (such as figures 21 and 23), it is the Cumulative Recall Percentage which is graphed.

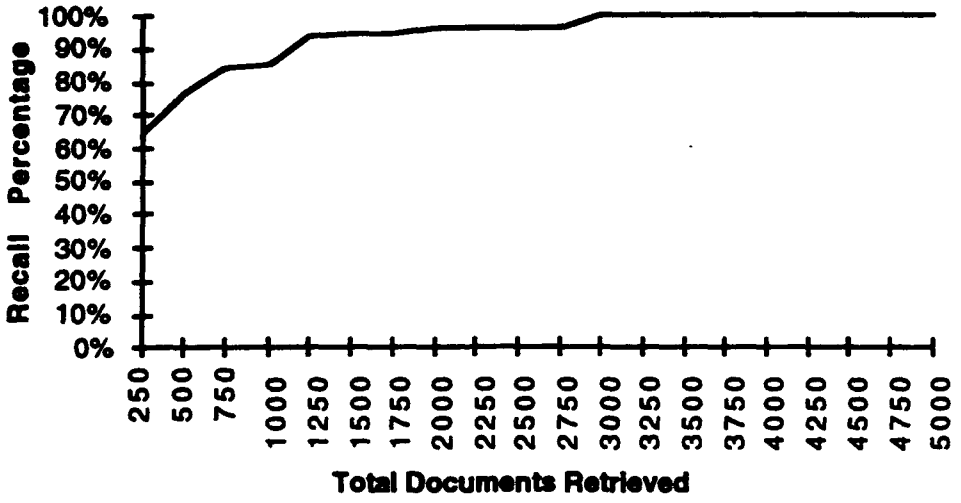


Figure 23 Cumulative Recall Percentage for Query #110

The above table and graph show the results for a single query in the TREC-2 query suite. When performing the final statistical analysis (next section), these tables and graphs are created for all queries. Then, the percentage values are averaged over all of the queries to give the final, average recall percentage graph.

3.3 Test Results

The following text retrieval tests were run:

1. Full TREC-2 topic, with and without dictionary
2. Only the topic description, with and without dictionary
3. Only the topic description, without the Thesaurus (but containing other dictionary sources)
4. Full TREC-2 topics, with and without grouping terms into sets

Each of these tests is described in the sub-sections which follow.

3.3.1 Full TREC-2 Topic Description, with and without Dictionary

The first test was to evaluate the utility of the ConQuest dictionary for the full TREC-2 topics. These are test runs where all of the words in the TREC-2 topic description from all sections of the topic are added to the query (except for those which were manually removed when the queries were generated).

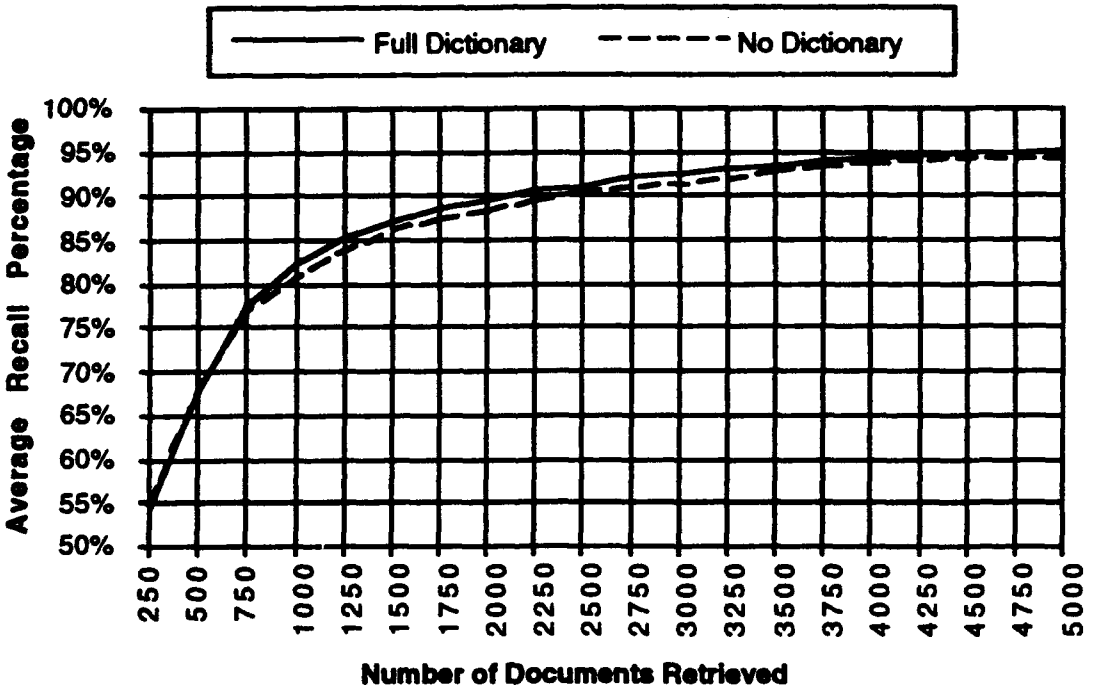


Figure 24 Full TREC-2 Topics, With and Without Dictionary

These results are encouraging. First, the non-dictionary version has definitely lower recall, if only one or two percentage points. Second, the relationship holds throughout the range (from 250 to 5000 documents) which improves the likelihood that this relationship is valid.

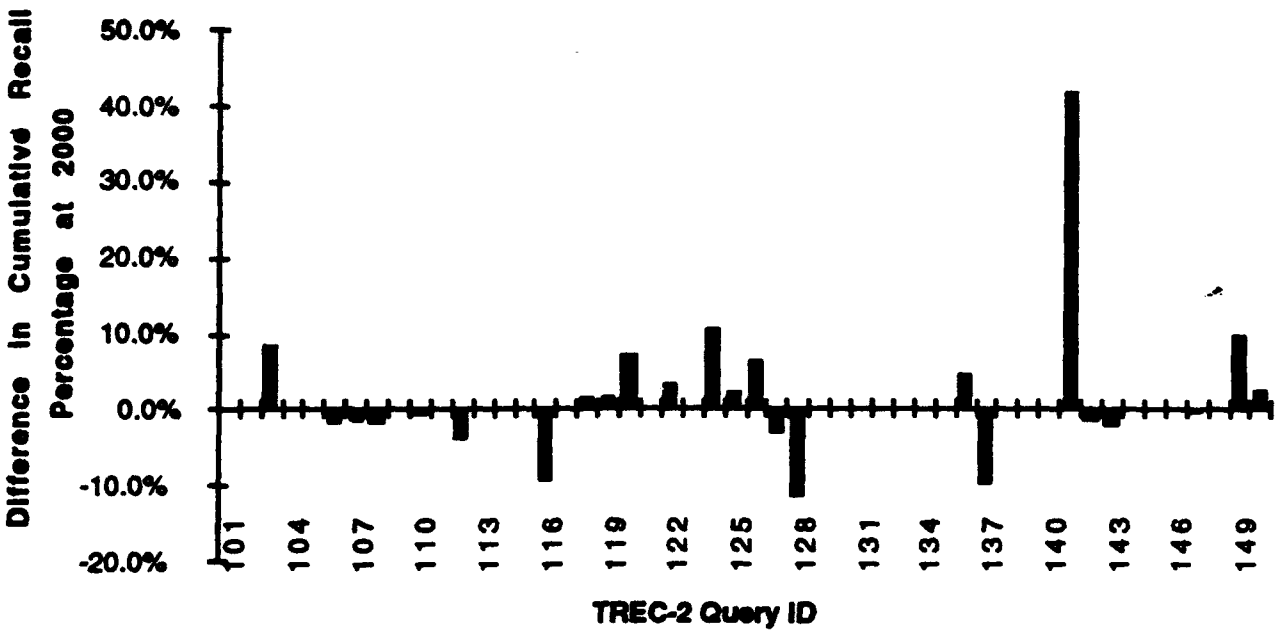


Figure 25 Difference in Cumulative Recall Percentage for Each Query at 2000

Figure 25 shows the difference in cumulative recall percentage for each query. This is done by taking the recall percentage for each query with the dictionary minus the percentage without the dictionary.

As you can see, most queries (26) had no change in accuracy due to the dictionary. An equal number of queries (12) showed an improvement as showed a decline.

However, one query (Query 141) showed a very large improvement (43%). This query was on the trade deficit between the United States and Japan. The drastic improvement was likely due to the addition of the words "Import" and "Export" to the query which otherwise did not have them. These words were automatically added by the dictionary.

This highlights the key improvement that dictionaries bring to a search system. If the user is unfamiliar with the domain over which he/she is searching, the ability of the dictionary to add the necessary key terms can dramatically improve the search accuracy of the system.

3.3.2 Reduced TREC-2 Topic Description, with and without Dictionary

The results from section 3.3.1 are not surprising, since similar results were reported at the TREC-2 conference. It was suggested at that time that additional runs, operating on a subset of the topic description, be executed.

This idea is appealing, because of how the original TREC-2 topic descriptions were generated. Each TREC-2 topic description contains quite a lot of good data for performing searches. Not only is there a long narrative (one or two paragraphs) describing the types of documents desired, but there is also a "concepts" section which gives additional terms which were added from known relevant documents (by the TREC-2 committee).

It is very unlikely that naive end-users would have the patience to enter such a long query description. Most users in the commercial market tend to enter small phrases, of just two or three words. The TREC-2 topic descriptions, on the other hand, were generated as if for a government intelligence agency (an equally valid end-user, but perhaps not as lucrative).

To run this next test, only words from the three shortest sections of the TREC-2 topic description were used. All other words were automatically removed. The three sections used were 1) The domain, 2) The topic title, and 3) The description.

The domain is usually a two or three word description of the general domain of the query. Very few domain words were actually used, because most were judged to be too general and were manually removed.

The topic title is a short phrase which describes the topic to be retrieved, but the particular aspects which are desired. It is usually much more specific than the domain.

Finally, the topic description is a single sentence which describes the types of documents desired. It usually describes the aspects of the topic which will be judged relevant.

```
<num> Number: 101
<dom> Domain: Science and Technology
<title> Topic: Design of the "Star Wars" Anti-missile Defense System
<desc> Description:
Document will provide information on the proposed configuration, components,
and technology of the U.S.'s "star wars" anti-missile defense system.
```

Figure 26 Example TREC-2 Topic with Domain, Title, and Description

The results of executing these reduced queries shows that the dictionary gives a much larger performance boost than before.

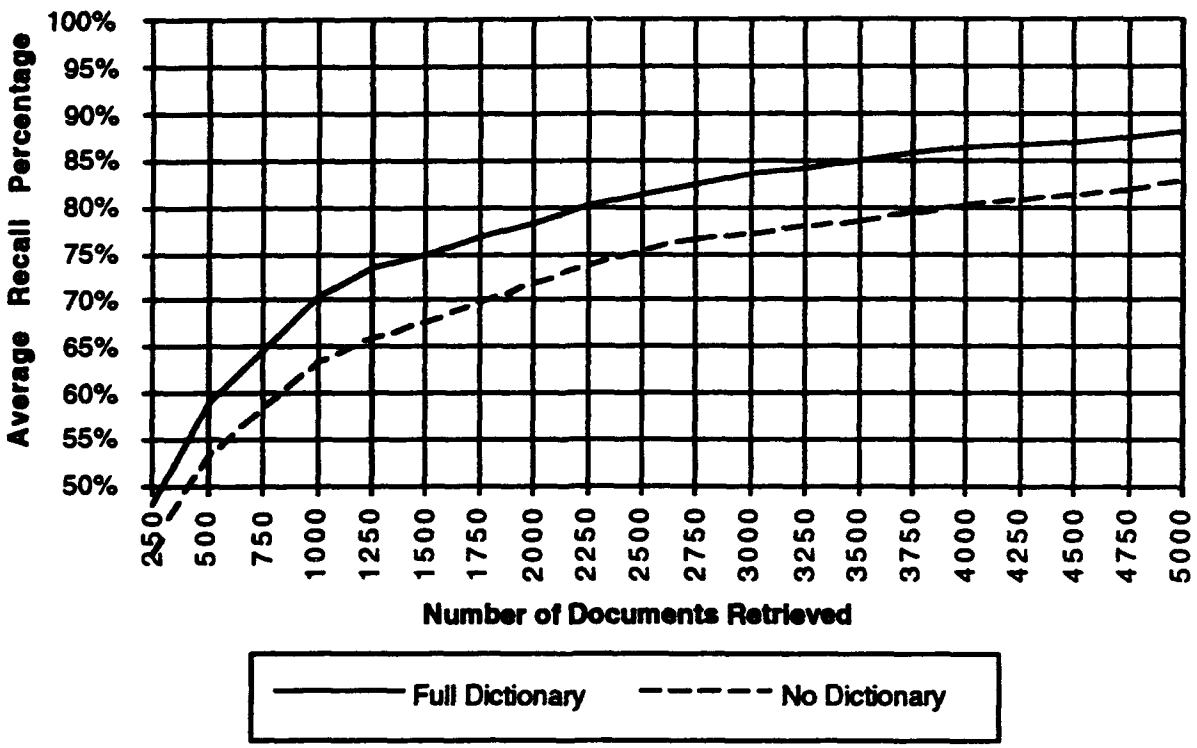


Figure 27 Example TREC-2 Topic with Domain, Title, and Description

With a much smaller query, the difference is now from 5 to 10 percent, rather than just 1 or 2 percent. It is apparent that smaller queries can gain more benefit from the use of dictionaries and semantic networks.

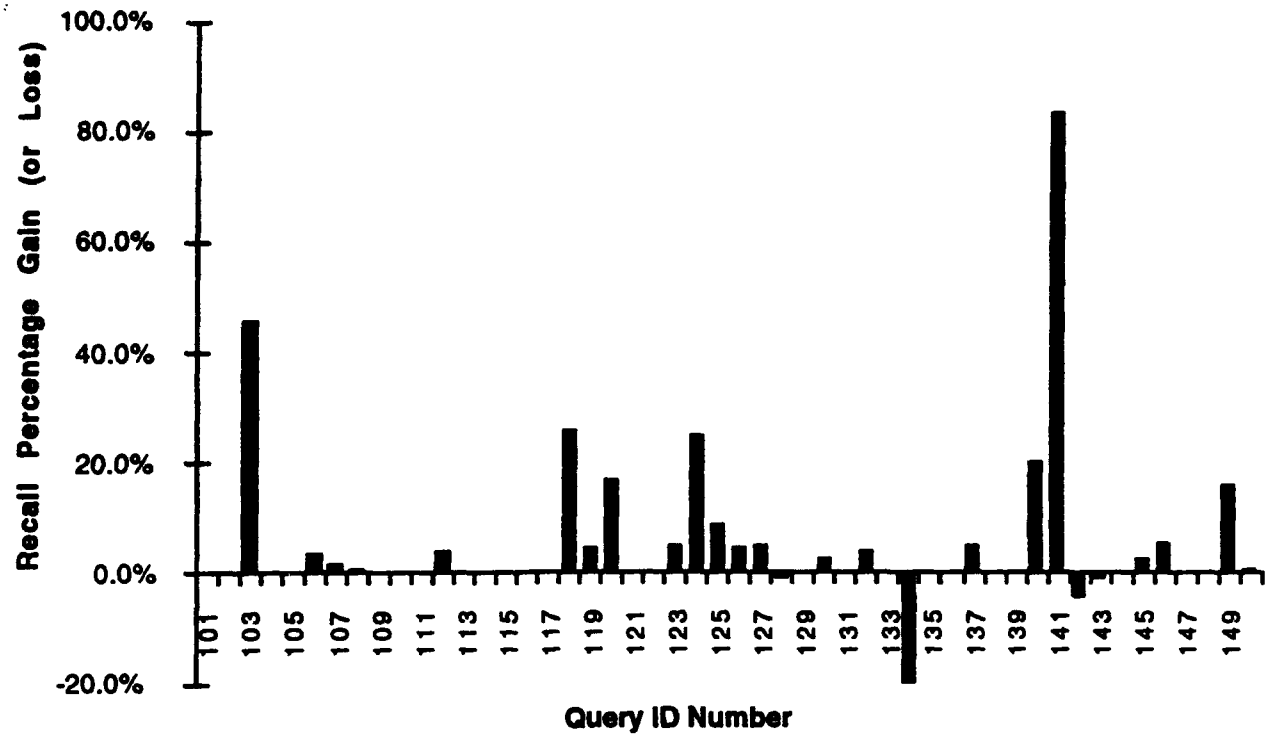


Figure 28 The Percent Difference Using a Dictionary for all TREC-2 Queries

The histogram also shows a dramatic improvement. While there are still many queries (24) which show no difference, there are now many more queries (21) which show an improvement with the dictionary and very few (4) which show a decline.

Also notice that query number 141 shows an even larger improvement before. Without a dictionary, the recall percentage of query 141 is just 8%. After automatically expanding words with the dictionary, the recall percentage jumps to 91%, an 83% improvement.

We would expect that the results would get even more and more dramatic as the queries get smaller and smaller. The queries for this test averaged from 10-20 words each. Very small user queries of just 2 or 3 words could be dramatically improved by the ConQuest dictionary expansion process.

3.3.3 Results Showing the Effects of Different Dictionaries

With the Auto-Query program, a test run can specify the dictionaries to be included in the test. Without a particular dictionary, the expansions for that dictionary will be removed from the query. In the following tests, the thesaurus was removed from the dictionary. This leaves WordNet as the only signification dictionary contributor.

These tests were run on the Domain, Title, and Description fields from the TREC-2 topic descriptions. The Narrative and Concepts sections were removed.

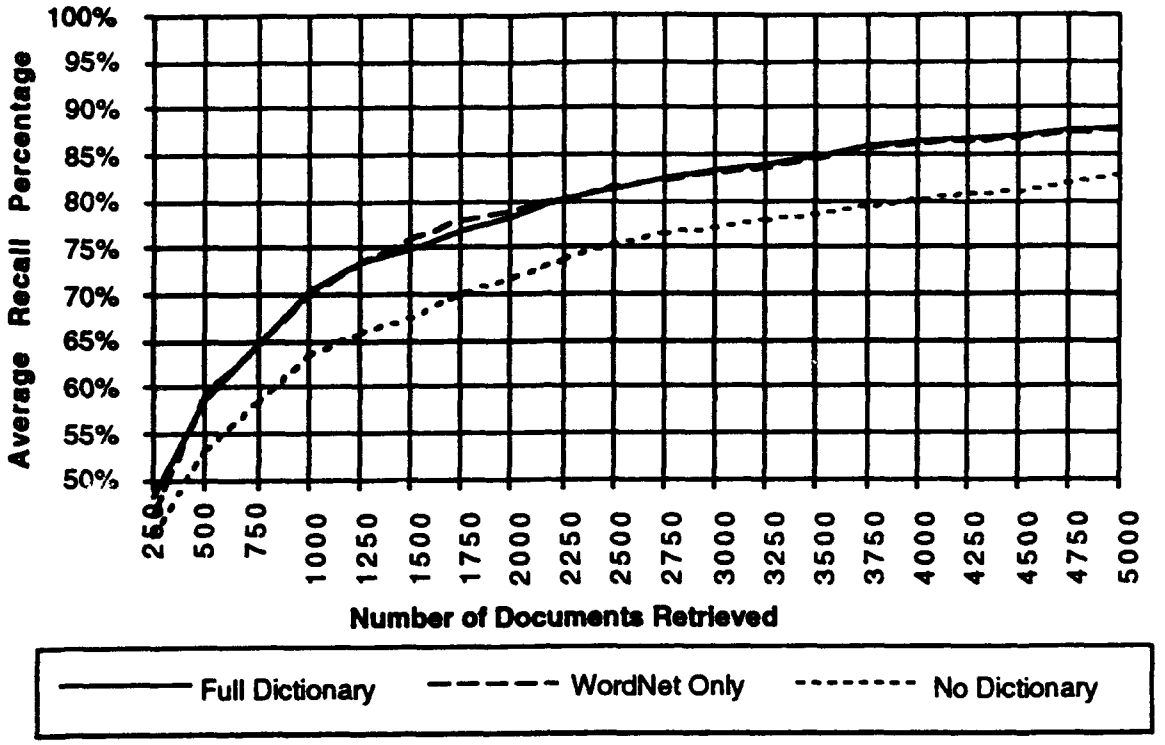


Figure 29 Results without the Thesaurus

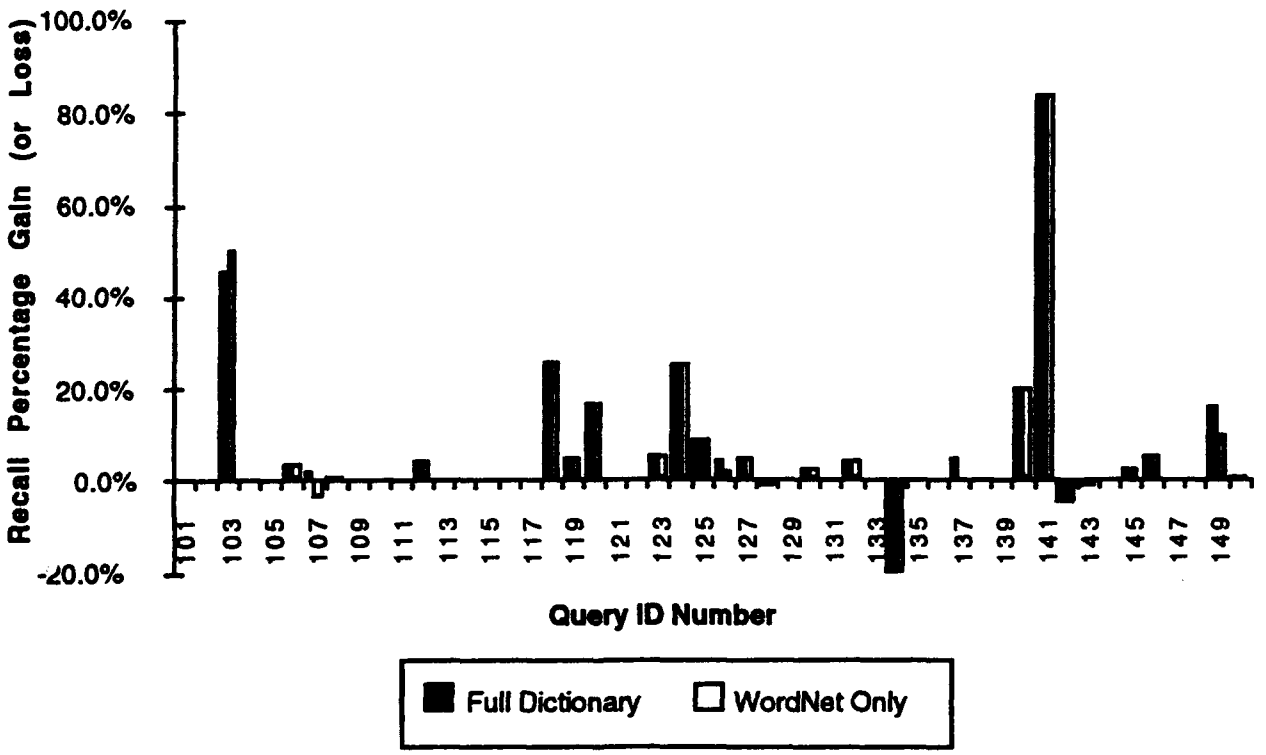


Figure 30 Changes in Recall Percentage for Two Different Dictionaries

These results were very surprising, because we expected a large difference due to removing the thesaurus from the ConQuest dictionary. However, there is practically no difference (and even an improvement in spots).

Without the thesaurus, the only other resource of consequence is WordNet. This implies that WordNet is primarily responsible for the performance improvements seen in the previous section.

A possible explanation of this result, after reviewing the strengths and weaknesses of the Thesaurus in section 2.2.3, is that verbs, adverbs, and adjectives do not play a significant role in queries in the TREC-2 descriptions. The thesaurus is very strong on these parts of speech (verbs, adverbs, and adjectives), and very weak in nouns. If it is the case the nouns are the best search terms, then this result makes sense. However, further study will be required to validate this hypothesis.

3.3.4 Full TREC-2 Topics, with and without Grouping Terms into Sets

The final test was to see if grouping terms into sets made any significant difference in the search results. All of the tests results shown above used the terms as grouped into conceptual sets. See section 3.1.3 for a detailed description.

Remember that words within sets are treated with the Accrue() function which is more "OR-like", and then the sets are combined with the Average() function, which is more "AND-like". It was assumed that this combination into sets would improve the performance of the system by reducing the number of times that many words from a single set would occur in a document while completely neglecting other sets.

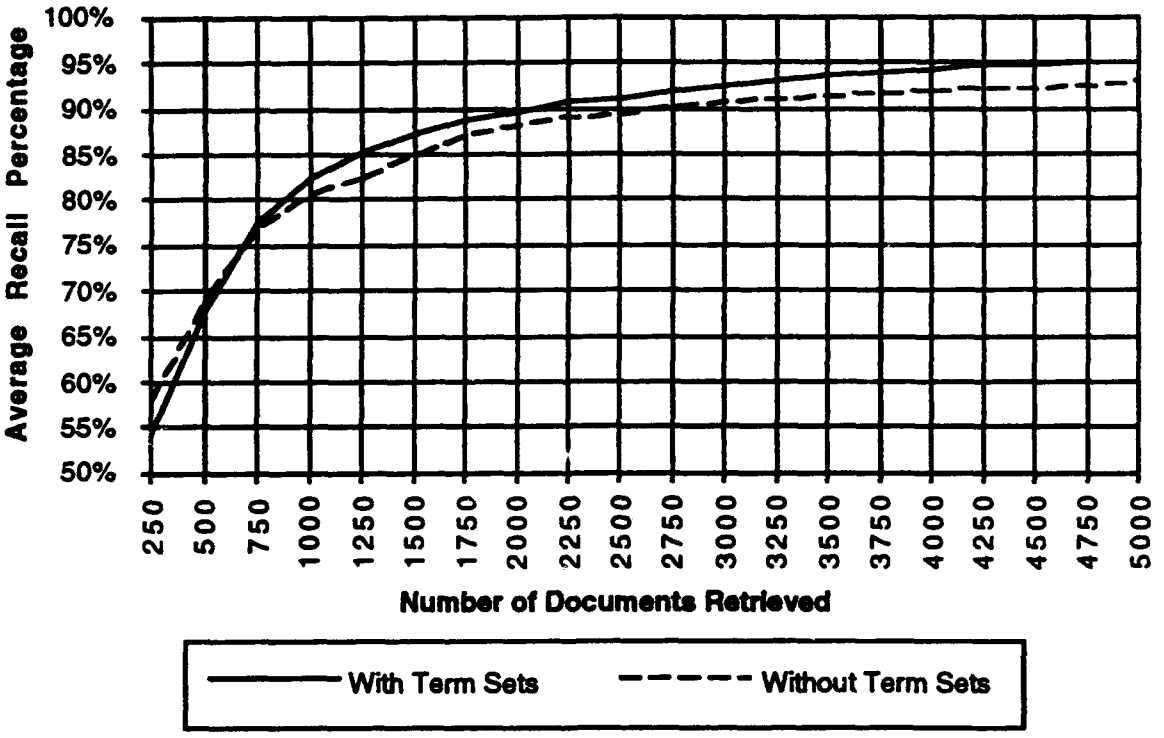


Figure 31 Recall Percentages with and without Grouping of Terms into Sets

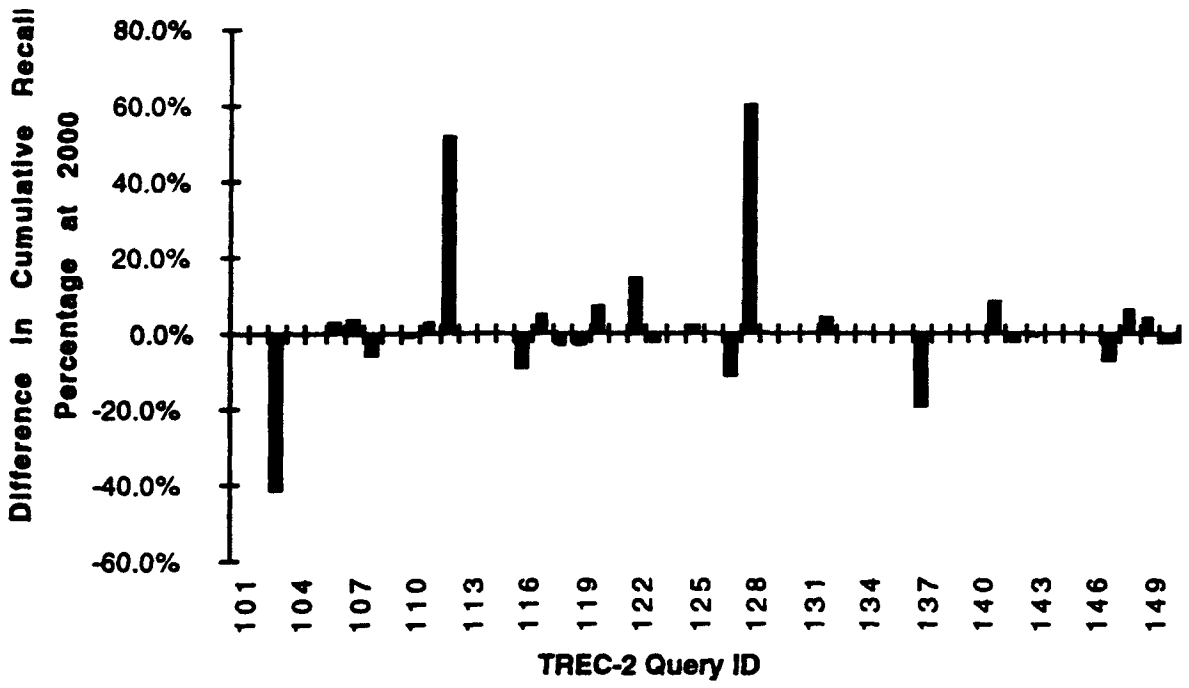


Figure 32 Differences in Recall Percentage for each Query

The differences in the two different algorithms are slight. An equal number of queries (13) showed improvements as declined, with two queries showing large gains and one showing a large loss. The average difference is about 2 or 3 percentage points in cumulative recall.

4 Other Text Retrieval Functions

The previous three sections describe the formal tests and evaluations that were run using ConQuest with dictionaries, especially WordNet. This section describes numerous other improvements and enhancements that have been studied. Some of these improvements have been tested, some implemented, and some just studied.

This discussion provides the natural prelude to the next phase of this contract. Many of these enhancements could be tested and evaluated thoroughly in Phase II.

4.1 Adding Terms to the Dictionaries

It was shown in section 3.3.2 that using dictionaries to augment queries does have a significant difference on query performance, especially for short queries. In addition, some queries can be improved significantly (as much as an 83% improvement) with dictionaries.

So a natural way to further improve the query performance is to broaden the coverage of the dictionaries, so that more different kinds of terms can be automatically processed. It would be best, of course, to add terms which are not already in the dictionary. In other words, the emphasis is to broaden the coverage, not necessarily deepen the coverage.

To this end, ConQuest has done some initial explorations and has forged some relationships for the following terms:

- **Business & financial terms** - ConQuest is working with a company called **SEC Online** which has a database of company names, ticker symbols, and industry codes. It is possible to use this database to automatically recognize company names and to expand the company

name to include terms related to the industry for which the company is most commonly associated. For example, ADM (Archer Daniels Midland) would be associated with farming and agricultural products.

- **Encyclopedia expansion** - Infonautics, an on-line search service to be introduced next year, has purchased ConQuest to make reference materials available on-line to high school students. Such materials include encyclopedias, dictionaries, the complete works of Shakespeare, Cliff Notes, etc.

  One useful form of term expansion is to apply a query to an encyclopedia as the first step towards searching a more general database. Once the main concepts are located in the encyclopedia, the encyclopedia entries can be used as the search over the general database. This idea and others will be pursued by ConQuest and Infonautics over the next year.

- **Domain Specific Terms** - When the multi-layered dictionary system is complete and commercially available, ConQuest will be able to fully exploit this technology to add domain specific terms (for example, technology and microelectronics) to the ConQuest dictionary structure.

  Special domain specific terms have been added for a special customer of ConQuest, namely the National Library of Agriculture. While only a few domain specific terms were added, we can identify that these terms do improve the performance of the search system, especially for naive searchers. If someone uses the system who is unfamiliar with the domain (such as Agriculture), then the domain specific term can automatically expand from the layman's term to industry specific terms. Further, the dictionaries can be helpful to choose meanings of words which can help guide the user in constructing accurate searches.

4.2 Query Processing

Many additional improvements have been studied under the general heading of "Query Processing". These improvements attempt to automatically improve the accuracy of the query, typically by extracting additional information from the query.

4.2.1 Automatic Word Sense Disambiguation

ConQuest has studied a mechanism for automatic word sense disambiguation using semantic networks and "domain analysis". The idea is to choose meanings of words which are most closely related to meanings of the surrounding words in the query.

This process can be accomplished with the following algorithm:

1. Expand all meanings of all words
2. For each word, go through all expanded terms for all meanings
3. For each expanded term, see if the term is also contained in the expansion of another meaning in some other word in the query.
4. If the term expansion is elsewhere in the query, increase the confidence factor of the associated meanings for both words.

For example, if the query contains the word "river bank", then all meanings of the word "bank" will be expanded. For example, the meaning "financial institution" would expand to "loan", "deposit", and "S&L", while the meaning for "shore" would expand to "ocean", "crick", and "river". So, in this simple example, the obvious choice for the meaning of "bank" is "shore", because that is the meaning which is most closely related to the other meanings in the query.

ConQuest has previously tested a system which performed automatic word meaning disambiguation using WordNet 1.1 on a database of the Wall Street Journal. At the time, the accuracy of the system was minimal: only about 10% of the words were correctly disambiguated based on the meanings and

term collocations. Also note that this disambiguation was performed on the words in the document, which has much more contextual information.

It has been determined that the primary stumbling block to a practical implementation is the size of the semantic networks. The WordNet 1.1 networks simply did not contain enough links to make this process feasible. And while WordNet 1.4 has many more terms, it is still not enough. Our study has indicated that a semantic network would need to be 10x to 100x larger than the current ConQuest dictionary to sufficiently increase the accuracy of automatic word sense disambiguation.

However, ConQuest has started exploring the possibility of automatically **extending** the existing semantic networks using term collocation statistics. Recent studies at AT&T have shown that large "concordances" (a large set of related terms) can be built using collections of related terms derived from Roget's Thesaurus. ConQuest has studied the possibility of using the same techniques to build concordances for all meanings of all words in the ConQuest dictionary.

The process for building a concordance for a word meaning is as follows:

1. Start with a word meaning
2. Expand the meaning using the ConQuest semantic networks
3. For each expanded term, find all occurrences of the term in a large database (such as TREC-2)
4. Histogram all words which occur in a 20-50 word neighborhood of the expanded term
5. Use statistical regressions to sort the neighboring terms by their likelihood of relevancy to the original expanded term
6. Collect these neighboring term lists for all expanded terms
7. Take the strongest intersection of all neighboring terms across all expanded terms in the meaning.
8. This intersection is the concordance for the word meaning

These techniques serve to bootstrap the dictionary from the existing semantic networks to a very large database of related terms. Our studies have shown that the dictionary could grow very large (as much as 1000x larger) using these techniques. Such a dictionary would have sufficient numbers of related terms for each meaning to make an accurate word sense disambiguation routine.

4.2.2 Term Weighting Based on Syntax

When analyzing the results of the "WordNet only" dictionary and the "Complete" dictionary, we surmised that the Webster's Thesaurus was less useful because it primarily contained expansions for verbs, adverbs, and adjectives in the English language.

This analysis, if it holds true, provides the first clues that perhaps nouns really do make better search terms than adjectives or verbs. If this is the case, the syntactic analysis could have a positive affect on the query performance.

The purpose of a syntactic analysis would be to determine the part of speech for every word in the query. This has two advantages: 1) the meanings that user has to select from could be restricted to meanings with a compatible part of speech, and 2) the nouns in the sentence could be weighted higher, since they make better search terms.

ConQuest has developed a commercial-quality syntactic chart parser that operates on text at 10 MB per hour on a 486 IBM-PC Clone computer (33 Mhz). These speeds are sufficiently fast to explore the possibility of syntactic analysis not only for queries, but also for indexes as well.

4.2.3 Negation in the query

Many of the TREC-2 queries have negation. As an example:

> Any reported changes to original design, or any research results which might lead to changes of constituent technologies, are also relevant documents. However, reports on political debate over the SDI, or arms control negotiations which might encompass the SDI, are NOT relevant to the science and technology focus of this topic, unless they provide specific information on design and technology.

Figure 33 Example of Negation in a TREC-2 Topic Description

Examples like this one show how complex negation can be. One would like for the words "political debate" and "arms control negotiations" to be negatively weighted, to reduce the likelihood that such documents will be retrieved.

Fortunately, the ConQuest chart parser (see section 4.2.4) can be used for syntactic analysis of this type. Specific rules can be generated to recognize negation, and to account for the specific ways in which items can be negated. Such processing could significantly improve the accuracy of a search system, especially for the TREC-2 queries. Studies by Carnegie Group have shown that negation is essential to achieving 90% + in overall system precision and recall.

To rank documents with precision, ConQuest would perform a query over the positive words in the query, and another query over the negative words. The two results would then be combined together to achieve the final relevancy score for the document. Such a two-pass mechanism avoids many algorithmic problems which could require iterative processing and convergence of a solution.

4.2.4 Numeric Range Processing

In the most recent release of the ConQuest commercial product (version 3.0), numeric range processing was introduced. This style of processing enables users to search using numeric ranges. Queries such as "100-200 employees" can be located in the full text of documents and can be accurately ranked and evaluated.

However, ConQuest has not yet evaluated the effectiveness of such queries. One possibility would be to find the TREC-1 and TREC-2 queries which specifically call out numeric ranges, and evaluate the performance of the system with numeric range checking on and off. Such tests are possible with the most recent release of ConQuest.

4.3 Ranking

Several additional ranking criterion have been studied by ConQuest, but (as noted in section 3.2.4) these studies still have a long way to go.

The ranking algorithms which could improve coarse grain rank include 1) Inverse Document Frequency, 2) Inference nets based on semantic networks, and 3) Threshold Analysis.

The algorithms which could improve fine grain rank include 1) Inverse Document Frequency, 2) Position within document, 3) Inference nets from semantic networks, 4) Ranking window size, and 5) threshold analysis.

The following is a brief description of each algorithm.

4.3.1 Inverse Document Frequency

"Document frequency" is a measure of the frequency of a term in the entire database of documents. The assumption is that very frequent terms are less reliable search terms and so should be weighted lower than others.

ConQuest has used statistical regressions of word statistics to show that there is a strong correlation between the inverse of the document frequency and the probability that a term will return relevant documents. However, we are currently unsure of how to use this information to adjust the weights of expanded terms. A more detailed analysis will be required.

4.3.2 Position within Document

The idea with document position is that some words in a document should count for more than other words. Words in the title, words which are in **bold** font, and words in an abstract should probably count more than the words in the body of the text.

With another customer, ConQuest has experimented with this idea. The database contained year 1986 from the Medline database of medical abstracts. The Medline database has been subject-coded manually with "MESH" terms (stands for "MEdical Subject Headings").

The test involved counting hits which occurred in the MESH fields as twice the strength of other hits in the document. For example, the terms in a query such as "Breast Cancer in Males" are all potential MESH terms. ConQuest searches for terms in the body of the text as well as in the MESH headings and increases the strength of hits in the MESH.

The results of this test appear to be dramatically improved over treating the terms in the MESH as normal search terms. However, the results at this point are only anecdotal. A true quantitative analysis has yet to be performed.

4.3.3 Derive Inference Net from Semantic Networks

In section 3.1.3 a method for grouping terms into sets was described. This can be done manually, or is performed automatically when a word is expanded to related terms using the ConQuest dictionaries.

This idea of creating sub-sets of terms is a small step towards a general "inference net" framework as described by Bruce Croft and Howard Turtle. The inference network in this case is the network by which terms are combined into sets (using the Accrue() function) and then the sets are combined for the entire query (using the Average() function).

More nesting of terms, and more exotic inference nets can be automatically generated from the ConQuest semantic networks. Portions of the semantic network could be abstracted into inference nets which then provide the specific ranking function for a query.

There are two ways this could be performed:

1. (easy) Use the collocations in WordNet. WordNet contains many descriptive phrases which are identified as words separated by underscores, such as "notify_formally". These terms are not strictly idioms, because their meaning as a whole is pretty much the same as their meaning separately.

   However, these collocations could provide another layer in the inference net (subsets of sets) which could be used to further improve the accuracy of the ranking algorithms.

   One possibility for using the collocations would be to use a new feature in ConQuest, that of "fuzzy phrases". These are phrases (such as the TREC collocations) which use proximity information in a pass/fail fashion. The words in the phrase are allowed to be out of order, or

to be separated by one or two irrelevant terms. If these conditions are violated, then none of the hits are counted.

2. (hard) Extract portions of the WordNet semantic networks as inference nets. For example, each related concept in WordNet could be represented as a node in the inference net. Documents which contain terms which are related to the concept will serve to strengthen the corresponding node in the inference net. These concepts can then be combined with a more AND-like function such as Average().

4.3.4 Threshold Analysis

One problem with creating more and deeper nested sets of terms (as suggested in the previous subsection) is that the functions currently in use (Accrue() and Average()) are linear. In other words, the results are the same no matter how the terms are combined into nested sets.

For example, the Average(Average(A,B), Average(C,D)) = Average(A, B, C, D), and the same is true of the Accrue() function. This is because the functions are essentially linear in nature.

This implies that new functions should used which could take advantage of the subset nesting. In order for nested sets to make a difference, they should be non-linear. Then, it would be possible to nest the terms to any extent and the results would make sense.

Drawing from the field of Neural Networks, a good function might be a rounded threshold function. Such a function could account for many of the various ranking formulae which have been used, even the boolean techniques.

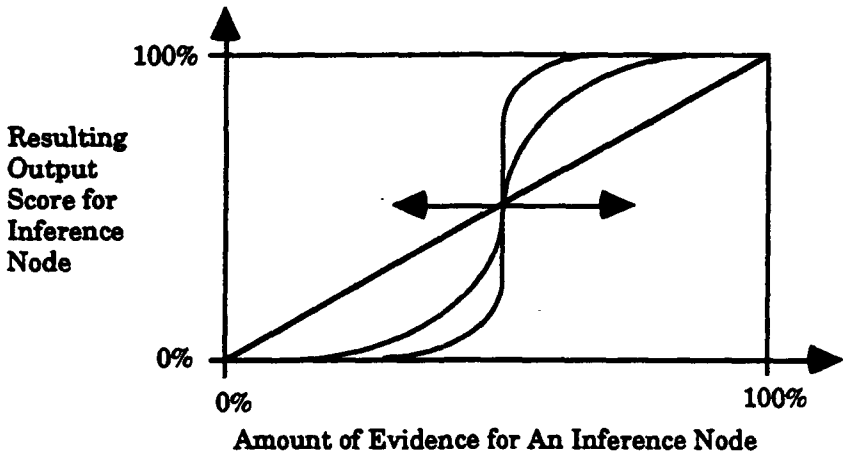


Figure 34 Possible Threshold Functions for Combining Evidence for an Inference Node

Such a function would have two parameters:

1. The threshold value - The point on the X-Axis where the threshold goes vertical
2. The "roundness" of the threshold - The amount of curvature in the threshold

By varying these parameters, the threshold function can mimic many different functions, including boolean OR and AND, the average() function, and a coarse approximation of the Accrue() function.

The results of testing with the threshold function could be plotted in a three dimensional graph. The X-Axis would be threshold value, the Y-Axis would be the roundess factor, and the Z-Axis would be the cumulative recall percentage. Such a graph would clearly show the optimum function for combining sets in more complicated inference nets.

4.3.5 Ranking Window Size

A final ranking algorithm which deserves exploration is adjustment to window size for fine-grain ranking. ConQuest currently uses a window size equal to 4 times the number of main concepts in a query. So, a query of 4 terms would pass a 16 word window over each retrieved document. The strength of the window is based on the number of hits which occur in the window. These window strengths are used to compute the "maximum hit strength" and "average hit strength" statistics identified in section 3.2.1.

However, for many TREC-2 queries, the Maximum hit value appears to be so low as to be useless to determine the probability of relevance for a document. Initial tests have shown that this is primarily because the window is too small. The text of documents vary so widely that such a small window eliminates too many good combinations.

A recent ranking algorithm prototype has increased the window to a fixed size of 200 words. Next, a bell-shaped weighting function is applied to the window for the terms which occur in the window. Terms near the center of the window are ranked at near 100%. Terms near the edges of the window will be ranked lower.

Such an algorithm appears to give better performance for the Maximum Hit and Average Hit functions. However, the evidence is currently anecdotal. Quantitative analyses are needed.

4.4 Evaluation

Finally, there are two different tests which could be performed to test different aspects of the ConQuest system. The first is the use of more statistical regressions, and the second is a link-by-link evaluation.

4.4.1 Statistical Regressions

Some initial analysis of the fine-grain statistics were shown in section 3.2.4. Some additional statistical tests could include the following:

1. Word Statistics for Coarse-Grain ranking
   a. Word frequency in query
   b. Inverse document frequency
   c. Adjusted term weights from dictionary expansion
   d. Number of meanings for the word from the dictionary
   e. Position in the query for the term
   f. Part of speech for the word derived from syntactic analysis

   The goal of this analysis would be to improve the weight of query terms using the statistics shown above. These new weights would be set such that strongly weighted terms would most likely retrieve relevant documents.

   Initial studies of this data by ConQuest have shown that results could be improved by as much as 10 to 15 percent.

2. Word Statistics for Fine-Grain ranking
   a. All the same statistics from test #1 above
   b. Frequency of the term in the document
   c. Proximity of the term to other terms in the document using the ranking windows described in section 4.3.5.
   d. Position of the term in the document (title, abstract, authors, etc.)

   This analysis would provide adjusted term weights based on the word and its usage within a document. These new term weights could be used as inputs to fine-grain ranking statistics.
3. Fine-Grain ranking statistics
   a. Maximum hit
   b. Average hit
   c. Coarse grain rank
   d. Number of hits
   e. Other functions?

   This analysis would provide a fine-grain ranking function to re-sort the document provided by the coarse-grain algorithm. Such a sorting should propagate the most relevant terms to the top of the retrieved list.

   Initial tests of this regression were shown in section 3.2.4.

4.4.2 Link-by-Link Evaluation

The same test as described in section 2.5 could be applied to full text retrieval experiments. The goal would be to test the usefulness of each link in the ConQuest dictionary in terms of actual precision and recall for retrieved documents (*as opposed to precision and recall of retrieved words*).

To perform the test, the same database of semantic link weight tables would be used. One table to test each different link type. Two sets of results would be required: one before traversing the link, and one after. The difference in the accuracy of these results would identify the true usefulness of the link.

A second mechanism for link-by-link evaluation would be to identify which expanded terms were derived by what link types. Then, calculations could produce the probability that a particular link type would add relevant documents to the retrieved document list (as opposed to the probability that it would add relevant terms to the expansion list.